Ph.D. DISSERTATION

Process Fault Detection and Diagnosis via the Integrated Use of Graphical Lasso and Markov Random Fields Learning & Inference

마르코프 랜덤 필드 학습 및 추론과 그래프 라쏘를 활용한 공정 이상 감지 및 진단 방법론

BY

CHANGSOO KIM

February 2019

DEPARTMENT OF CHEMICAL & BIOLOGICAL ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Process Fault Detection and Diagnosis via the Integrated Use of Graphical Lasso and Markov Random Fields Learning & Inference

마르코프 랜덤 필드 학습 및 추론과 그래프 라쏘를
활용한 공정 이상 감지 및 진단 방법론

지도교수 이 원 보
이 논문을 공학박사 학위논문으로 제출함

2019년 2월

서울대학교 대학원

화학생물공학부

김 창 수

김창수의 공학박사 학위 논문을 인준함

2019년 2월

위 원 장: _____
부위원장: _____
위 원: _____
위 원: _____
위 원: _____

Abstract

Fault detection and diagnosis (FDD) is an essential part of safe plant operation. Fault detection refers to the process of detecting the occurrence of a fault quickly and accurately, and representative methods include the use of principal component analysis (PCA), and autoencoders (AE). Fault diagnosis is the process of isolating the root cause node of the fault, then determining the fault propagation path to identify the characteristic of the fault. Among the various methods, data-driven methods are the most widely-used, due to their applicability and good performance compared to analytical and knowledge-based methods. Although many studies have been conducted regarding FDD, no methodology for conducting every step of FDD exists, where the fault is effectively detected and diagnosed. Moreover, existing methods have limited applicability and show limited performance. Previous fault detection methods show loss of variable characteristics in dimensionality reduction methods and have large computational loads, leading to poor performance for complex faults. Likewise, preceding fault diagnosis methods show inaccurate fault isolation results, and biased fault propagation path analysis as a consequence of implementing knowledge-based characteristics for construction of digraphs of process variable relationships. Thus a comprehensive methodology for FDD which shows good performance for complex faults and variable relationships, is required.

In this study, an efficient and effective comprehensive FDD methodology based

on Markov random fields (MRF) modelling is proposed. MRFs provide an effective means for modelling complex variable relationships, and allows efficient computation of marginal probability of the process variables, leading to good performance regarding FDD.

First, a fault detection framework for process variables, integrating the MRF modelling and structure learning with iterative graphical lasso is proposed. Graphical lasso is an algorithm for learning the structure of MRFs, and is applicable to large variable sets since it approximates the MRF structure by assuming the relationships between variables to be Gaussian. By iteratively applying the graphical lasso to monitored variables, the variable set is subdivided into smaller groups, and consequently the computational cost of MRF inference is mitigated allowing efficient fault detection. After variable groups are obtained through iterative graphical lasso, they are subject to the MRF monitoring framework that is proposed in this work. The framework obtains the monitoring statistics by calculating the probability density of the variable groups through kernel density estimation, and the monitoring limits are obtained separately for each group by using a false alarm rate of 5%.

Second, a fault isolation and propagation path analysis methodology is proposed, where the conditional marginal probability of each variable is computed via inference, then is used to calculate the conditional contribution of individual variables during the occurrence of a fault. Using the kernel belief propagation (KBP) algorithm, which is an algorithm for learning and inferencing MRFs comprising continuous variables, the

parameters of MRF are trained using normal process data, then the individual conditional contribution of each variable is calculated for every sample of the fault process data. By analyzing the magnitude and reaction speed of the conditional contribution of individual variables, the root fault node can be isolated and the fault propagation path can be determined effectively.

Finally, the proposed methodology is verified by applying it to the well-known Tennessee Eastman process (TEP) model. Since the TEP has been used as a benchmark process over the past years for verifying various FDD methods, it serves the purpose of performance comparison. Also, since it consists of multiple units and has complex variable relationships such as recycle loops, it is suitable for verifying the performance of the proposed methodology. Application results show that the proposed methodology performs better compared to state-of-the-art FDD algorithms, in terms of both fault detection and diagnosis. Fault detection results showed that all 28 faults designed inside the TEP model were detected with a fault detection accuracy of over 95%, which is higher than any other previously proposed fault detection method. Also, the method showed good fault isolation and propagation path analysis results, where the root-cause node for every fault was detected correctly, and the characteristics of the initiated faults were identified through fault propagation path analysis.

Keywords: Process monitoring, Fault detection and diagnosis, Markov random fields modelling, graphical lasso, kernel belief propagation

Student Number: 2014-21514

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Research Motivation

Process monitoring is an essential part of safe and stable process operation. Since various forms of process faults can occur during the operation of a process, it is vital to set up effective control structures for stabilizing the process when a fault is initiated. However, some faults, which may alter the operating conditions of the process, cannot be maintained using control structures, and thus efficient process monitoring schemes have to be applied to detect these faults, and to diagnose the root cause of the fault and its characteristics so that proper actions can be taken to mitigate the changes that occur as a consequence the fault. The framework of using process control and process monitoring to safely operate process plants, and the components of process monitoring, are shown in Figure 1.1. Since there is no fixed terminology regarding the various sections of process monitoring [9], the terms fault detection and diagnosis (FDD) will

be used throughout this study, where fault diagnosis includes the steps of fault node isolation and propagation path analysis.

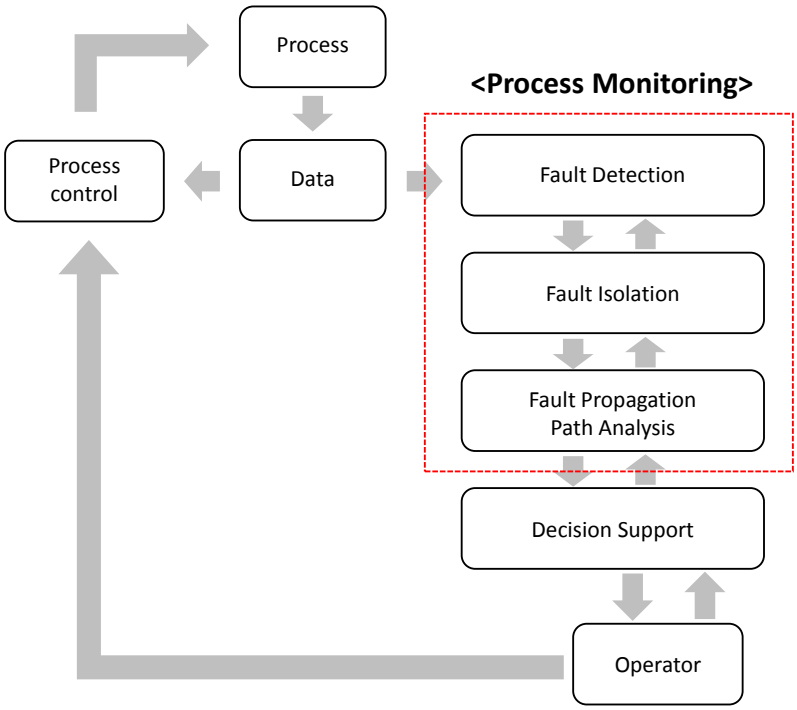


Figure 1.1: Measures for stable and safe operation of process plants, and the role of process monitoring.

Despite its importance, no comprehensive methodology for effectively detecting and diagnosing the fault exist, due to various difficulties of application. One of the main bottlenecks of applying process monitoring methods is the large number of monitored variables within a process plant, and the computational burden induced from the numerous variables.

To statistically monitor multivariate processes, data-driven monitoring methods are widely studied compared to analytical and knowledge-based monitoring methods. One of the reasons for this is that data-driven monitoring methods are better applicable since large amount of data can be obtained from process operations. On the other hand, building plant-wide analytical models is infeasible, and process knowledge is difficult to acquire before developing the process monitoring scheme and may bias the monitoring results should inappropriate process knowledge be applied. Conventional data-driven monitoring methodologies include the use of dimensionality reduction methods, such as the use of principal component analysis (PCA), independent component analysis (ICA), and Fisher discriminant analysis. Using linear feature extraction techniques, these methods reduce the large dimensions of variables into a few number of components, and develops fault detection statistics such as T^2 and SPE values, for multivariate fault detection. Also, the root-cause variable can be detected by computing the contribution value of each variable with respect to the T^2 and SPE statistics. While these dimensionality reduction methods are easy to apply and have good computational efficiency, they have intrinsic limitations that restrict their application. One of the main limitations is

the linearity of the reduced dimensions. Since many process variables have nonlinear, non-Gaussian forms, they cannot be modelled using linear feature extraction methods. Moreover, the characteristics lost in the process of dimensionality reduction results in deduced performance of the monitoring methods. Although the feature extraction methods select the optimal number of components, individual properties of the variables are lost when they are projected onto the extracted components. This leads to poor performance of FDD accuracy and detection speed. Thus monitoring methods which retain the variable properties and is able to preserve the nonlinear, non-Gaussian characteristics is required.

Recent studies tend to focus on either of the two aspects of process monitoring, fault detection or fault diagnosis. Also, they implement various data processing and variable modelling methodologies into the field of process monitoring, to overcome the limitations inherent in variable projection methods. These methods include the use of autoencoders, sparse global-local preserving projections, and Bayesian networks. A number of studies apply supervised learning algorithms such as neural networks and support vector machines for monitoring, but these methods are not considered here since they require a priori knowledge of process faults for training the models.

Use of autoencoders for process monitoring show good fault detection accuracy, as shown in the work by Yan et al. [3]. Using variant autoencoders including denoising autoencoders and contractive autoencoders, the activation functions of intermediate layers effectively capture the nonlinear characteristics of monitored variables. Application of

the variant autoencoder based monitoring method shows improved monitoring results compared to conventional methods. However, the method is not able to diagnose the fault since it lacks measures of variable contribution with respect to the occurring fault. Also the fact that it is also a dimensionality reduction method limits its performance for faults occurring in complicated forms.

Global-local preserving projections is an enhanced version of the conventional dimensionality reduction methods, where the feature extraction process and the analyzing process of monitoring statistics are improved to become better applicable to process monitoring. The work by Bao et al. [10], Luo et al. [11], and Luo et al. [12] are such studies, where the global-local preserving projections method is implemented to effectively capture nonlinear characteristics of process variables, and to obtain sparse forms of principal components. Also, they apply variable-wise and group-wise contribution analysis, which allow extensive fault diagnosis. Whereas these methods show good performance in fault diagnosis, they still show mediocre performance regarding fault detection, due to the limitations of dimensionality reduction methods.

The use of Bayesian networks, and other forms of directed graphs, are another recent trend of process monitoring. By using process data and process knowledge on the sequence of process units, the graphical structure of the monitored variables are built, then process monitoring is performed based on the statistics obtained by graphical inference. Studies regarding directed graphical model monitoring include work by Verron et al. [13], Mori et al. [14], and Gonzalez et al. [2]. However, these methods

are limited by the fact that process monitoring is conducted based on fixed models of conditional relationships. Since monitored variables are connected by directed graphs, fault propagation occurring in opposite directions cannot be detected, leading to biased fault diagnosis. Also, the fact that Bayesian networks are incapable of presenting cyclic relationships, and the fact that conditional probability for a node in Bayesian networks do not take into account the intricate relationships existing between variables other than that of the parent, hinders good process monitoring performance.

Other methods focusing on fault diagnosis have been proposed as well. Such studies include the use of the Granger causality and the transfer entropy. Granger causality is a method for computing the causal relationship between two variables, calculating the influence of one variable to another using vector autoregressive models. Transfer entropy is a generalized form of Granger causality, where the mutual information term used in information theory is converted into transfer entropy by computing the conditional probability between two variables. While these methods can avoid biased fault diagnosis, different from fault diagnosis of Bayesian networks, it has certain limitations. First the computational load of transfer entropy is enormous, which inhibits its application to large processes. Also, the conditional relationships calculated from Granger causality and transfer entropy are bivariate relationships, which does not consider the effect of neighboring variables. Thus the variable contribution cannot be computed with credibility. Thus a process monitoring methodology with less computational load, and better expressibility of variable relationships is required.

1.2 Research Objectives

The objective of this thesis is to propose a novel comprehensive methodology for FDD, which is capable of effectively detecting various forms of complicated faults, and can preserve the characteristics of the variables so that variable relationships are efficiently modelled, leading to good performance in fault isolation and propagation path analysis. The proposed methodology comprises modelling the monitored variables into Markov random fields, which is an undirected probabilistic graphical model (PGM), and the learning and inference of Markov random fields to develop fault detection statistics and contribution analysis values.

First, a novel variable modelling technique using MRF, is proposed. The use of MRF allows extensive modelling of variable relationships including cyclic relationships. Next, the structure of the monitored variables is obtained by using the graphical lasso algorithm. Graphical lasso is a structure learning method for MRFs, which drives the parameters representing irrelevant variable relationships to zero, forming the MRF into a sparse structure. This allows efficient computation of the MRF inference process. Afterwards, the MRF is trained based on kernel density estimation (KDE) using normal process data, then used to calculate fault monitoring statistics regarding faulty process data. Finally, using kernel belief propagation (KBP), the MRF is trained in a nonparametric form different from the case of using KDE, and the conditional contribution values are evaluated to analyze the root-cause node of the fault, and to determine the fault propagation path. To verify the performance, the proposed methodology is applied

to the widely used Tennessee Eastman process (TEP). The TEP model is a benchmark process used for evaluting various fault monitoring methods, thus serving the purpose of performance comparison.

1.3 Outline of the Thesis

The outline of the thesis is as follows. In Chapter 2, the MRF modelling and structure learning methodology for process variables are introduced, with details regarding the graphical lasso algorithm and optimal structure learning. Also, the proposed method is applied to the TEP, and the results are given. In Chapter 3, the fault detection section of the proposed methodology is described in detail, and application results to the TEP are given. In Chapter 4, the fault diagnosis section of the proposed methodology is described in detail, and its application results to the TEP model are given. Chapter 5 presents the conclusion and suggestions for future studies.

Chapter 2

Markov Random Fields Modelling, Graphical Lasso, and Optimal Structure Learning

2.1 Introduction

Graphical models in general are visual realizations of variable relationships. They consist of nodes and the edges connecting each of the nodes, where the nodes represent the random variables and the edges represent the conditional or parallel relationships between variables. Graphical models are based on the concept of declarative representation, where the model for the system of interest is constructed, and reasoning of various situations is carried out using the knowledge built inside the model [15].

Combined with the probability theory, probabilistic graphical models (PGM) provide a formal framework for considering multiple possible outcomes and computing the likelihood of each outcome. To compute the probabilities using PGMs, first the

graphical structure of the random variables and their edges, and the parameters that define the graphical model, have to be learned using training data. Then using the trained graphical model, the probability of the individual variables, or the entire group of variables, are computed based on the new observations, or test data. The first step of model training is called the learning process, and the second step of probability computation is called the inference process. Various algorithms of learning and inference for PGMs allow effective modelling and reasoning of random variables, such as the variational inference and junction tree algorithm [16]. Due to the intuitional representation capabilities and efficient probability computing algorithms, PGMs are used in various fields of studies, such speech recognition, gene regulatory data modelling, and protein modelling [15, 16].

Various efforts have been made to apply the variable modelling capabilities of PGMs into the field of process systems engineering. Since conditional relationships can be directly modelled, most of these efforts have been focused on causal modelling of process variables. Such studies include the ontology-based reasoning of process faults [17], Bayesian network based fault detection and diagnosis (FDD) [18–20], and HAZOP automation systems based on various forms of graphical models [21, 22]. Catergorically for FDD, Bayesian networks were mainly used, such as the studies by Verron et al. [13], Mori et al. [14], Gonzalez et al. [2], Dey and Stori [18], Yan and Yao [19], and Verron et al. [20]. While these methods provide an effective means of fault detection, and provide the basis for intuitive fault isolation and propagation path

analysis, inherent limitations of the Bayesian network deteriorate the performance of these methods. Whereas undirected graphical models may be a solution to these obstacles, there has been no study regarding them since the modelling and computation of undirected graphical models are much more complicated compared to Bayesian networks.

In this chapter, the use of undirected graphical models, or Markov random fields (MRF), for the purpose of FDD is introduced, where the basics of MRF modelling and inference is explained in detail. Moreover, a novel algorithm for variable selection and structure learning of the MRFs for process variables is proposed, and the results of applying the method to the Tennessee Eastman process (TEP) is shown. The proposed method, namely the iterative graphical lasso, allows the variables with high correlations to be grouped together, while simultaneously providing the undirected graphical structure of the grouped variables. The proposed method and the results provided in this chapter become the foundation for applying the inference algorithms of MRFs to FDD, which will be introduced in Chapters 3 and 4.

2.2 Markov Random Fields

PGMs are used in various fields to model the relationship between variables, and to construct a model for learning the likelihood of variables with respect to the data set of interest [15]. After the variables are modelled into a graphical model, the parameters of the specific model are learned using training data, then the marginal probability of each

variable is obtained by using the observations, or test data, to inference the probability.

PGMs largely consist of directed and undirected forms. Directed acyclic graphical models, or Bayesian networks, are effective in expressing the conditional relationships among variables, and are used in various fields of study, such as variable causality analysis, protein sequence modeling, and speech recognition [15, 16]. While the Bayesian network can intuitively represent conditional dependencies among variables, it is limited in expressing certain forms of relationships. For instance, when trying to model the two conditional dependencies, $A \perp C|B, D$ and $B \perp D|A, C$, of the four variables A, B, C, and D, the only feasible representation is by using an undirected graph, as shown in Figure 2.1. It is not possible to represent both conditional dependencies using directed graphs, showing that they have limited representation capabilities. Also, it is not possible to model cyclic relationships with Bayesian networks, since by definition the marginal probability of a directed graph having cyclic relationships is infeasible to compute.

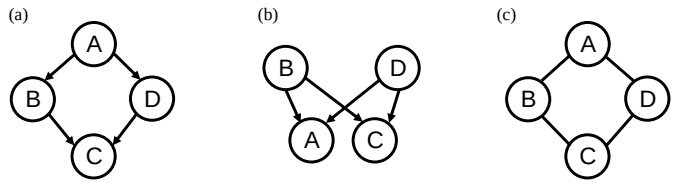


Figure 2.1: Limited expressibility of directed graphical models.

As a result, certain systems, such as process plants, cannot be expressed thoroughly using Bayesian networks due to the abundance of cyclic connections such as in reflux flows. In these occasions, the use of undirected graphical models are more appropriate. Although the learning and inference process of undirected graphical models are computationally more expensive compared to Bayesian networks, they provide extensive modelling capability. Undirected graphical models model the system of interest using potentials, which are factors that represent the characteristic of a node, or multiple nodes. This can be represented as shown in Equation 2.1:

$$f(x) = \frac{1}{Z} \prod_{C} \psi_C(x_C) \tag{2.1}$$

Functions $\psi_C(\cdot)$ are the factors for the designated set of variables, and C is the set of all groups. The variable Z is the partition function, or the global normalization factor, for normalizing the effect of all potentials, and is defined as shown in Equation 2.2:

$$Z = \sum_{x} \prod_{C} \psi_C(x_C) \tag{2.2}$$

When the potentials consisting the undirected graphical model are positive, the undirected graphical model is called a Markov random field (MRF). MRFs are used in various fields of study, such as image segmentation and computer vision, gene regulatory network prediction, and spatial financial interaction analysis [15, 16]. The benefit of MRFs compared to Bayesian networks is that it has extensive expressibility of variable relationships, and that it is able to consider the effect of all of the variables

within the system during inference, owing to the existence of the partition function.

In a system where the variables are modelled as MRFs, the main interest would be to calculate the marginal probability of the node of interest, with respect to a specific observation. This process is called inference, and is only possible after learning the parameters of the MRF. Thus the main tasks when dealing with MRFs are the computation of the partition function Z, parameter learning of the potentials comprising the MRF, and efficient computation of the inference process. The most widely used parameter learning and inference algorithm for MRFs is the belief propagation algorithm, where the conditional probability of one node affecting another is modelled into a message, and then all of the messages from one node to another are iteratively calculated until convergence. However, the computational load of most belief propagation algorithms are very expensive, owing to the partition function and the number of neighboring nodes for a specific variable. To mitigate the computational load and make the belief propagation process feasible, it is important to know the structure of MRFs, making the graph structure sparse so that only the essential relationships of variables are left. But likewise as the case in belief propagation, learning the structure of a MRF is computationally infeasible due to the existence of the partition function, and most applications of MRFs have knowledge of the MRF structure beforehand.

2.3 Graphical Lasso

The lasso is a widely used regression method first proposed by Tibshirani [23], which makes use of the l_1 norm to set irrelevant coefficients to zero while calculating regression parameters. Coefficients having a value of zero means that the certain variable can be excluded from the regression model, and thus the lasso works as a variable selection methodology [24]. Recently the lasso has been extended to be applied in graphical models, namely graphical lasso, allowing the core subset of a graphical model to be found efficiently. The graphical lasso was first proposed by Friedman et al.[25], and various methods for applying the lasso on graphical models have been proposed recently[26–34], along with research making use of the sparse graphical models induced from the use of the graphical lasso[35]. The graphical lasso is one of the few structure learning algorithms for MRFs, when the MRF is modelled into a pairwise MRF. Following the work proposed by Friedman et al.[25], the graphical lasso problem can be expressed as:

$$\max_{\boldsymbol{\Theta}} \log\det(\boldsymbol{\Theta}) - tr(\mathbf{S}\boldsymbol{\Theta}) - \rho\|\boldsymbol{\Theta}\|_1 \tag{2.3}$$

where $\boldsymbol{\Theta}$ is the precision matrix, or the inverse of the covariance matrix of the variables $\boldsymbol{\Sigma}$, $\mathbf{S}$ is the empirical covariance matrix, and ρ is the regularization parameter for the l_1 norm. Eq. 2.3 is the Gaussian log-likelihood of the data, partially maximized with respect to the mean parameter, μ. Maximizing Eq. 2.3 is equivalent to solving the graphical lasso problem, and Friedman et al.[25] solves this problem by first partitioning

the matrices $\mathbf{W}$, and $\mathbf{S}$ as:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W_{11}} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix}, S = \begin{pmatrix} \mathbf{S_{11}} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix} \tag{2.4}$$

then using the fact that the solution for w_{12} satisfies:

$$w_{12} = \arg\min_{y} \{ y^T \mathbf{W_{11}}^{-1} y : \|y - s_{12}\|_\infty \leq \rho \} \tag{2.5}$$

and that the above equation is equivalent to solving the dual problem,

$$\min_{\beta} \frac{1}{2} \|\mathbf{W_{11}}^{1/2} \beta - b\|^2 + \rho \|\beta\|_1 \tag{2.6}$$

where $b = \mathbf{W_{11}}^{-1/2} s_{12}$. Using the relationship that $\mathbf{W\Theta} = \mathbf{I}$, the sub-gradient equation for maximization of the log-likelihood Eq. 2.3 is:

$$\mathbf{W} - \mathbf{S} - \rho \cdot \mathbf{\Gamma} = 0 \tag{2.7}$$

Thus by iteratively solving and updating the lasso problem Eq. 2.7, the graphical lasso problem can be solved to return the essential subset of a given graph. As a result, a sparse precision matrix ($\mathbf{\Theta}$) is obtained, where the zero values in the matrix denote the absence of an edge between the two nodes. This way, the structure of an undirected graphical model can be obtained, in a sparse form so that the computational complexity of inferencing the graph is mitigated. Since the graphical lasso problem assumes that the pairwise MRFs are modelled as Gaussian graphical models, meaning that the pairwise

variables can be expressed using only Gaussians, it should be carefully applied when dealing with variables having high nonlinearity, or non-Gaussian features. The results obtained when implementing the graphical lasso on nonlinear variables may not be an optimal structure, considering the assumption implied when first modelling the MRF. However, it still provides an approximate form of the graph structure, which can be used in situations where identifying the members of a MRF is more important than obtaining the exact information of the edges connecting these members.

2.4 MRF Modelling & Structure Learning

2.4.1 MRF modelling in process systems

When modelling process variables into MRFs, it is inappropriate to use the clique notation shown in Equation 2.1, since it requires the exact edge structure of the MRF for modelling. The structure and number of cliques greatly affect the computational load and accuracy of probability calculation, and thus it is infeasible to use clique based MRF formulations when the exact structure of MRF is uncertain. Also, many of the existing belief propagation algorithms are designed to be used for pairwise MRFs. Thus process variables are modelled into pairwise MRFs using node potentials and edge potentials, as shown in Equation 2.8:

$$P(X) = \frac{1}{Z} \prod_{(s,t)\in E} \psi_{st}(X_s, X_t) \prod_{s\in V} \psi_s(X_s) \tag{2.8}$$

Analogous to the clique based formulation, the partition function Z can be expressed as in Equation 2.9.

$$Z = \sum_{x} \prod_{(s,t)\in E} \psi_{st}(X_s, X_t) \prod_{s\in V} \psi_s(X_s) \tag{2.9}$$

When modelling process variables into MRFs, it is not possible to know the model structure beforehand, since they have complex and diverse relationships. Thus they are first modelled into a fully connected graph, then formulated into a more exact structure by applying the graphical lasso algorithm.

2.4.2 Structure learning using iterative graphical lasso

As shown in the previous section, the graphical lasso algorithm may be used for approximate structure learning of process variables, since the process variables have nonlinear and non-Gaussian properties. However in many cases, the resulting structure of one application of the graphical lasso is not a graph with eliminated edges from the fully-connected version, but rather certain variables with low correlations tend to be cut-off from the whole structure. Thus the results of applying the graphical lasso shows a form with only the core variables of the MRF grouped together, rather than preserving the original set of variables with only certain edges eliminated. This is not a desired form of MRF, since for process monitoring purposes every variable should be within a group of variables, so that multivariate statistical monitoring is possible. To this end, the iterative graphical lasso algorithm is proposed and applied in this study, where the

stranded variables from a single run of the graphical lasso algorithm are applied the algorithm one more time, so that another set of variables that are highly correlated, can be produced. This process is repeated until every process variable is affiliated within a group. The workflow of the iterative graphical lasso algorithm is shown in Figure 2.2.

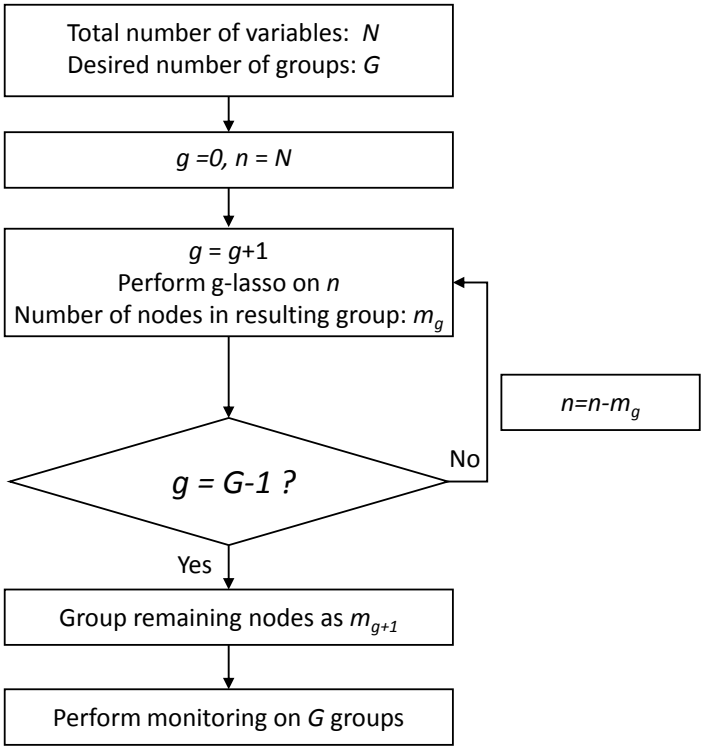


Figure 2.2: Iterative graphical lasso algorithm for grouping the variables into relevant relationships.

An issue arises when applying the iterative graphical lasso algorithm to process variables, regarding the regularization parameter ρ. The structure of the MRFs and the members comprising them can have large alternations with different values of ρ, for each iteration. Thus it is important to select the value of ρ so that the desired number of groups are met, with each group showing optimal regression results. When using the lasso regression, some methods for optimally selecting the value of ρ are used, such as the Alternating Direction Method of Multipliers (ADMM) proposed by Boyd et al. [36]. However, the objective of graphical lasso is different from lasso, since its main purpose is to find the interrelationships among variables, as proposed in Friedman et al. [25]. Selection of the parameter ρ is discussed in Banerjee et al. [33] and Meinshausen and Bühlmann [31], where the appropriate selection of ρ is discussed with respect to appropriate neighbor selection of the nodes. Neighbor selection is a former version of the graphical lasso, but the focus of the discussion is on selecting ρ so that pairings of nodes with low correlation are avoided, and also on the consistent formation of graph structures. These points are less relevant when using the graphical lasso, since the graphical lasso solves a dual optimization problem which is guaranteed to provide consistent graph structures among the nodes. The only concern when using the graphical lasso is the convergence of the covariance calculation algorithm, since solving the dual optimization problem guarantees consistent selection of neighboring nodes. Thus in this study, rather than using an algorithm for optimal values of ρ, the desired number of groups are set prior to the application of graphical lasso, then the values of ρ for each

iteration of the graphical lasso are determined so that the desired number of groups can be appropriately obtained with evenly distributed number of variables for each of the groups. As is the case in this study, the selection of ρ is dependent on the specific problem of interest, and different methods are used in various studies, such as in Liu et al. [37], and Menendez et al. [38].

2.5 Application of Iterative Graphical Lasso on the TEP

The iterative graphical lasso procedure introduced in the previous section is applied to the TEP model.

First, the number of desired groups are determined through sensitivity analysis of the fault detection accuracy (FDA) for the different choices of groups numbers, which is a performance evaluation metric discussed in detail in Section 3.3. In the current application, the cases of using 1, 2, or 3 groups are excluded from analysis, since dividing the variables into less than 4 groups degrades the benefits of Glasso-MRF monitoring regarding efficient fault detection and fault propagation analysis. To compare the performances of the different cases, the average FDA for each setting is calculated with respect to the 28 faults of the TEP. When dividing the variables into the specified number of groups, the value of ρ is tuned so that each group bears approximately the same number of variables. The resulting groups with the variables included in each group, and the average FDA obtained by applying different group numbers are shown in Table 2.1.

Table 2.1: Average fault detection sensitivity results for use of different number of groups.

No. of Groups	Group members	Average FDA
4	G1: 1, 7, 10, 11, 13, 16, 18, 40, 42, 43, 44, 47, 48 G2: 4, 20, 21, 22, 29, 30, 32, 34, 36, 45, 49, 51 G3: 2, 3, 23, 24, 25, 27, 28, 31, 33, 37, 38, 41 G4: 5, 6, 8, 9, 12, 14, 15, 17, 19, 26, 35, 39, 52	98.96%
5	G1: 1, 7, 10, 13, 16, 42, 43, 44, 47 G2: 11, 18, 20, 21, 22, 28, 48, 49, 51 G3: 3, 4, 25, 30, 32, 33, 34, 35, 36, 40, 45 G4: 23, 24, 26, 27, 29, 31, 37, 38, 39, 41 G5: 2, 5, 6, 8, 9, 12, 14, 15, 17, 19, 52	98.16%
6	G1: 1, 10, 13, 16, 42, 43, 44, 47	95.33%

	G2: 7, 11, 18, 20, 21, 22, 28, 45, 48, 49	
	G3: 3, 9, 30, 32, 34, 35, 36, 40, 51	
	G4: 23, 24, 25, 27, 29, 31, 33, 38	
	G5: 2, 4, 5, 6, 15, 26, 37, 39, 41	
	G6: 8, 12, 14, 17, 19, 52	
	G1: 1, 10, 42, 43, 44, 47	
	G2: 7, 11, 13, 16, 18, 20, 22, 48	
	G3: 3, 4, 21, 32, 36, 40, 45, 51	
7	G4: 23, 28, 29, 30, 34, 49	87.7%
	G5: 24, 25, 27, 31, 33, 38	
	G6: 2, 6, 26, 35, 37, 39, 41	
	G7: 5, 8, 9, 12, 14, 15, 17, 19, 52	

Results from Table 2.1 show a definite trend in the change of FDA with regards to the number of groups: the average FDA decreases with the increasing number of MRF groups. This is an expected result, since dividing the variables into too many groups results in more loss of inter-variable correlation when learning and inferencing the graphical model. Another trend is that the gradient of increase in average FDA tends to decrease when the number of groups is decreased. Thus the optimal number of groups has to be selected by considering the trade-off between the average FDA value, and the computational cost of MRF learning and inference during the process of monitoring. Comparing the cases using four and five groups, when dividing the monitored variables into 4 groups, the average FDA increases by a small amount of 0.8%, compared to the case of using five groups. Although the increase in average FDA is consistent with the results of the cases of using six and seven groups, the improvement in monitoring performance is trivial compared to the additional computational complexity when learning and inferencing MRFs with more variables in each group (Table 3.3). Also, the fault propagation path analysis results that can be obtained by analyzing the fault detection time difference among the groups, become less significant when using fewer number of groups. Considering the sensitivity analysis results with respect to the number of groups, the most efficient form of monitoring the TEP is to divide it into five groups, since it shows good FDA values, while being computationally feasible and providing significant results from fault propagation analysis. Thus the number of appropriate groups for diving the TEP is determined to be five.

After the number of groups is determined, the iterative graphical lasso is applied. Since it is decided that 5 groups are to be made, the regularization parameter ρ is tuned so that each group bears 9 to 11 variables. The iterative graphical lasso process, the value of ρ for each run and the variables within each group, is shown in Figure 2.3. The value of ρ decreases with each iteration, since the number of variables decrease and a smaller value of ρ is sufficient to acquire the desired sparsity. It can be seen that the iterative graphical lasso algorithm shows reasonable results, since the variables consisting each group are inter-correlated according to the variable attributes, the sequence of the process units, and the associated control loops. Group 1 consists of input and output flowrates (1, 10) and their corresponding manipulated variables (42, 43, 44), and the unit pressure variables (7, 13, 16), and their corresponding manipulated variables (47). Group 2 consists mainly of temperature (11, 18, 21, 22) and manipulated variables (48, 49, 51) nodes. The manipulated variables are connected to the temperature variables through the control loop, and the compressor work variable (20), and the composition of F (28) are naturally connected to the temperature variables through thermodynamic correlations. Group 3 consists of the input flowrates (3, 4) and its manipulated variable (45), and the compositions of the corresponding streams (25, 30, 32-36, 40). Group 4 consists of the composition variables of the input stream (23, 24, 26, 27), the purge stream (29, 31), and the product stream (37, 38, 39, 41). Group 5 consists mainly of flowrate variables (2, 5, 6, 14, 17, 19), and the level variables (8, 12, 15) that are directly related to them. The temperature variable (9) is related to the reactor level variable

(8) since they both represent reactor variables, and the manipulated variable node (52) relates to the temperature variable since it controls the cooling water flow. This analysis shows that all of the variable members of each group are correlated according to the three criteria, variable attributes, sequence of process units, and the associated control loops. The variable IDs and graph structure for each of the resulting groups, are saved to be used for the MRF monitoring process.

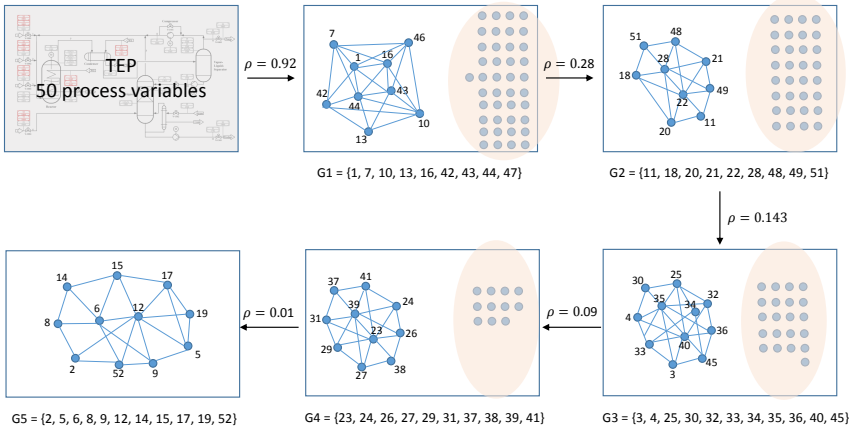


Figure 2.3: Results of applying the iterative graphical lasso on the monitored variables of the TEP. Numberings on nodes correspond to the variable numbers.

Chapter 3

Efficient Process Monitoring via the Integrated Use of Markov Random Fields Learning and the Graphical Lasso

3.1 Introduction

Process monitoring, or fault detection and isolation (FDI), is an essential part of product quality management and safe plant operation. There exist various methods for FDI, which can be categorized into three approaches, namely the analytical, knowledge-based, and data-driven methods[9]. Of these methods, data-driven methods are the most widely used in the industry, since they do not require a priori knowledge of faults and show good performance in terms of speed and accuracy. Data-driven methods make use of historical data to develop monitoring statistics, providing limits to the operation of process variables and detecting a data point as a fault when the monitoring limit is vio-

lated. Conventional data-driven monitoring methods include the use of dimensionality reduction techniques such as principal component analysis (PCA) [39–41], partial least squares (PLS) [42], and Fisher discriminant analysis [42]. Although widely used, the data-driven methods have certain aspects that limit their usage in terms of FDI. PCA and PLS have limitations in dealing with nonlinearity of the data, since they reduce the dimensionality of the variables upon the assumption of linearity. Also, it is difficult to isolate a fault using these methods, since the individual characteristics of the variables are simplified into reduced dimensions. Various studies suggest different dimensionality reduction techniques to resolve this problem, such as the use of kernel PCA, independent component analysis (ICA) [43, 44], and autoencoders. Kernel PCA incorporates the use of kernel space when reducing the dimensionality of the variable space, so that nonlinear relationships between variables can be considered more rigorously within the kernel space [45–47]. ICA [43, 44] makes use of statistically independent components that contain higher-order statistical information between variables. Autoencoders are a recently developed unsupervised machine learning technique, where a nonlinear activation function is used within encoders and decoders to effectively extract useful features from multiple variables [3]. All of these are useful in developing more appropriate monitoring statistics for nonlinear process variables, and succeed in enhancing the fault detection probabilities compared to conventional methods. However, they are not able to effectively isolate the fault, or provide a fault propagation path.

Apart from the studies that apply different dimensionality reduction techniques

to deal with nonlinearity of the variables, other studies suggest the use of sparse dimensionality reduction techniques or knowledge-based methods to overcome fault isolation problems. Sparse forms of dimensionality reduction allow better understanding of the relationships between variables, leading to good performance in fault isolation. Studies by Gajjar et al. [48], and Luo et al. [11] make use of sparse PCA for FDI. More complex forms of sparse dimensionality reduction techniques are incorporated as well. These studies include the work by Luo et al.[12], where process knowledge is used to construct a knowledge-based sparse projection matrix, and the work by Bao et al.[10], where sparse global-local preserving projections are used for FDI. While these methods are better suited for fault isolation compared to conventional dimensionality reduction techniques, they show only little improvement in fault detection accuracy, owing to the fact that they share the limitations of conventional methods, such as linear dimensionality reduction.

Studies on knowledge-based FDI include the use of neural-networks, support vector machines (SVM), and the Bayesian network. Neural-networks and SVMs have become more useful in the recent years owing to the boost in computational power, and studies such as the work by Chiang et al. [49] make use of these methods to achieve good monitoring performance. Bayesian networks have recently been used in FDI, such as in studies by Verron et al. [13], Verron et al. [20], Gonzalez et al. [2], and Zeng et al. [50]. Bayesian networks model the process to be monitored into directed graphical models, and use the causal relationships among variables to effectively isolate the fault and

analyze the fault propagation path. While these studies have shown great progress in the terms of FDI, they have a critical limitation in that, they require a priori knowledge of the process faults or the relationships between variables to work. Neural-networks and SVMs require data points classified as "faults" to train the monitoring classifier, which is impossible to obtain before a fault actually occurs. The directed edges of Bayesian networks have to be modeled beforehand to detect the cause of the fault upon occurrence, but this is difficult as well, since calculation of causal relationships are difficult in large-scale systems. Also, Bayesian networks have limited expressibility, such as not being able to model cyclic relationships of variables, whereas there are numerous cyclic relationships within chemical processes. These limitations restrain these methods from actually being applied in the industry. Thus a novel monitoring scheme that improves the monitoring performance of state-of-the-art monitoring methods, and also is capable of analyzing propagation paths, is required.

In this chapter, a monitoring methodology which is capable of dealing with process nonlinearity and fault propagation path anaylsis, while retaining good process monitoring performance in terms of fault detection speed and accuracy, is proposed. The method, denoted as Glasso-MRF monitoring throughout the paper, integrates the use of the graphical lasso and Markov random fields (MRF) modeling, to efficiently perform process monitoring. Also, Glasso-MRF monitoring allows fault propagation analysis [51, 52], to provide information on the characteristics of the fault and to prevent the fault from further affecting the process. This is an important feature of Glasso-MRF

monitoring, since fault propagation path analysis is an important step for identifying the fault and minimizing process damage, as mentioned in Chiang et al. [9]. To prove the performance of the proposed methodology, the well-known Tennessee Eastman Process (TEP) is used, to test out the monitoring performance using the proposed algorithm and compare the results to state-of-the art monitoring techniques.

The outline of chapter 3 is as follows. The preliminaries required for explaining the methodology is introduced in Section 2, then the Glasso-MRF monitoring methodology, along with its two subparts, the graphical lasso and MRFs, is explained extensively in Section 3. The implementation results of the method to the TEP is given in Section 4, along with the discussion of results and comparison of monitoring performance with other monitoring algorithms. Section 5 concludes the paper and explains the related future work.

3.2 MRF Monitoring Integrated with Graphical Lasso

The proposed monitoring scheme works in two steps: first the entire set of variables are grouped into smaller sets via the iterative graphical lasso algorithm, where the structure of the MRF is obtained as well. Then using the probability density calculation procedure of MRF, the monitoring limits of each group are calculated from the significance level α of the normal process data. During the monitoring process, the data points to be monitored are converted into probability values using the trained MRF model, then evaluated against the monitoring limits to determine whether it is in normal or

faulty condition.

3.2.1 Step 1: Iterative graphical lasso

The graphical lasso serves two purposes in the proposed monitoring scheme. One is the grouping of monitoring variables into inter-correlated sets as a sparse graph structure, and the other is mitigation of computational complexity of MRF learning and inference. By grouping the highly correlated variables, the time delay of fault detection is reduced, fault detection accuracy (FDA) is increased, and since the time it takes to detect the fault varies for different groups, it has the potential for fault propagation path analysis. The mitigation of computational complexity is a very important aspect as well, since the accuracy and speed of MRF learning and inference depends greatly on the number of nodes and edges.

Graphical lasso usually results in one subset of nodes with all of the other nodes eliminated from the group, which is not the desired form to be used in the MRF monitoring scheme. To effectively monitor the process, all of the variables should be included within a certain group so that they can undergo multivariate monitoring. To this end, an algorithm for iteratively grouping all of the process variables is proposed. The iterative graphical lasso algorithm is explained in detail in Section 2.5.

3.2.2 Step 2: MRF monitoring

To use MRF for process monitoring purposes, the parameters of the graphical model have to be trained, then a monitoring statistic obtained from normal process data has

to be calculated. As mentioned in the preliminaries section, training and inference of the MRF is computationally expensive, so an efficient method for calculating the probabilities of input data has to be used. In this study, kernel density estimation (KDE), a non-parametric method estimating the probability density function of graphical models, is used to train and infer the probability of the MRFs. KDE allows effective modelling of the nonlinear, non-Gaussian variables, and allows quick inference even though the training process is slow and computationally expensive. The calculation of a unknown probability density function using KDE is given in Eq. 3.1,

$$f_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{3.1}$$

where the variable f is the density of interest that is to be estimated, K is the kernel that is to be used to estimate the density function, and h is the bandwidth of the specific kernel. The most widely used kernel function is the Gaussian kernel, which is expressed as:

$$K_i(x, h) = \frac{1}{h\sqrt{2\pi}} \exp(\frac{-1}{2} \frac{\|x - x_i\|^2}{h^2}) \tag{3.2}$$

The Gaussian kernel is used throughout this study as well. Bandwidth is an important factor in kernel density estimation, having a great influence on the estimation results. To speed up the calculation process, an empirical equation for calculating bandwidth,

the Silverman's rule of thumb[53] is used to select a value for the bandwidth:

$$h = \frac{4}{n(d+2)}^{\frac{2}{d+4}} \cdot \mathbf{S} \tag{3.3}$$

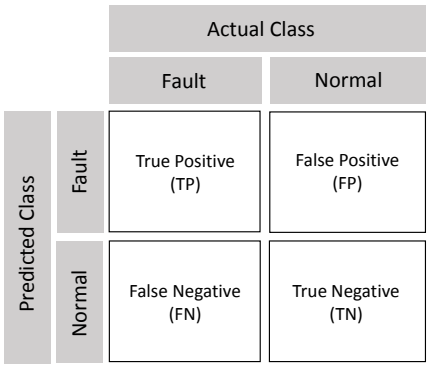
where $\mathbf{S}$ is the sampled covariance matrix of the dataset.

The overall process of MRF monitoring is, to first train separate KDE models for each of the MRFs, then obtain the monitoring limit using the trained KDE models, by setting the α confidence level of the probability values. For the new data points that are subject to monitoring, the probability of each point is inferred using the trained KDE model for each group of MRF, then their values are compared to the monitoring limits. If the inferred values are above the limit the data point is in a normal condition, whereas if below the value, it is in a faulty condition.

3.3 Implementation of Glasso-MRF monitoring to the Tennessee Eastman process

The proposed monitoring method is applied to the widely used TEP model, to test out its performance. Since most of the recent monitoring methods have been readily applied to the TEP, it serves the purpose of performance comparison. Two widely used metrics, fault detection accuracy (FDA) and fault detection rate (FDR), are used to evaluate the performance of the Glasso-MRF monitoring method. FDA and FDR are defined based on the binary classification test of the data points, as shown in Figure 3.1.

The TEP is explained in the following section, then the results of applying the proposed method in terms of FDA, rate and speed are shown, in comparison with conventional, as well as state-of-the-art monitoring methods.



- **Fault Detection Accuracy**

$$\text{FDA} = \frac{TP + TN}{TP + FP + FN + TN}$$

- **Fault Detection Rate**

$$\text{FDR} = \frac{TP}{TP + FN}$$

Figure 3.1: Binary classification of monitored data points (left), and the definitions of FDA and FDR[7] (right).

3.3.1 Tennessee Eastman process

The Tennesse Eastman Process (TEP) is a widely-used benchmark chemical process. After being first published by Downs and Vogel[1], it has been used to test out various control structures, optimization methods, and to verify the usefulness of different process monitoring techniques. It was first published in the Fortran language, but to enhance compatibility and usage, it was revised in various languages and certain modifications have been made to it. The most recent version is the MATLAB version published by Bathelt et al. [8], which resolved the numerical solver issues in the original version and comprises of three different control strategy models. Compared to the version in Downs and Vogel[1], this model implements the control strategy described in Ricker[54], and has 8 more additionally programmed faults, making up a total of 28 fault cases. Most of the previous studies use the dataset extracted from the Downs and Vogel[1] model, but since the MATLAB version shows more consistent numerical results compared to the original Fortran model, this version is used in this study. The 28 faults are tested out to verify the performance of the Glasso-MRF monitoring framework, then the first 21 faults are used to compare its performance with previously proposed monitoring methods.

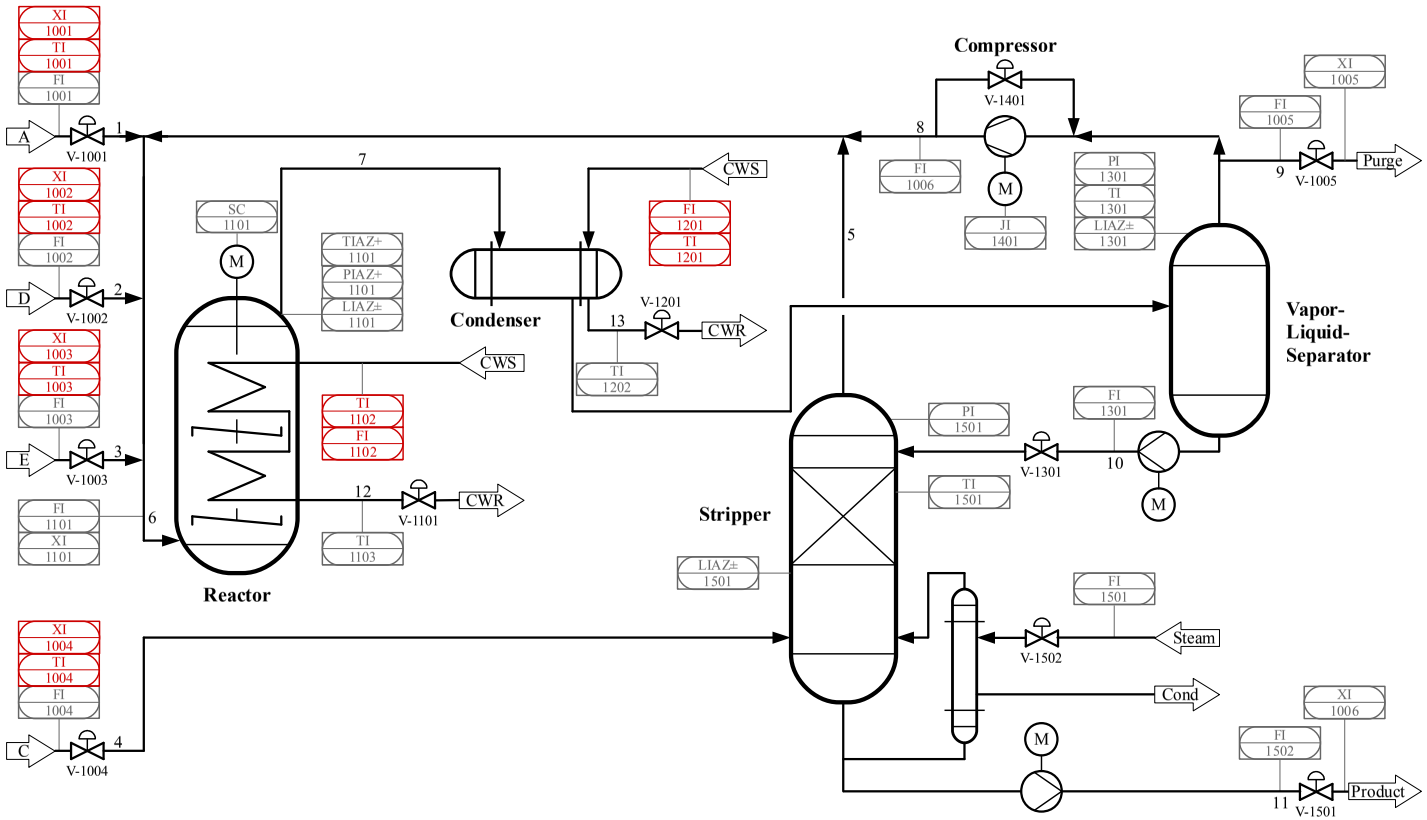


Figure 3.2: P&ID of the TEP model. Revised MATLAB version (Bathelt et al.[8]). IFAC Copyright is acknowledged.

The configuration of the TEP is given in Figure 3.2. The TEP model consists of five process units, the reactor, condenser, compressor, vapor/liquid separator, and the stripper. Four types of gaseous materials, A, C, D, and E, are put into the process to produce two types of products G and H, along with a by-product F. The data obtained from the TEP simulation consists of 22 continuously measured variables, 19 composition variables, and 12 manipulated variables. For monitoring purposes, only 50 of the 53 variables are used, as listed in Table 3.1. This is due to the fact that variables number 46, 50, and 53, which are the manipulated variables for compressor recycle valve, the stripper steam valve, and the agitator speed, respectively, retain constant values under the control strategy implemented and thus may degrade the performance of monitoring schemes. The set of 28 programmed fault cases are listed in Table 3.2. It should be noted that the description of fault 21 is different from Bathelt et al. [8], and is consistent with previous studies that monitor the TEP [3, 4, 10].

Table 3.1: Monitored variables in the TEP model. Numberings are based on the original paper (Downs and Vogel[1]).

No.	Process variables	No.	Process variables
1	A feed (stream 1)	18	Stripper temperature
2	D feed (stream 2)	19	Stripper steam flow
3	E feed (stream 3)	20	Compressor work
4	A & C feed (stream 4)	21	Reactor cooling water outlet temperature
5	Recycle flow (stream 8)	22	Separator cooling water outlet temperature
6	Reactor feed rate (stream 6)	23-28	Components A, B, C, D, E, F in stream 6
7	Reactor pressure	29-36	Components A, B, C, D, E, F, G, H in stream 9
8	Reactor level	37-41	Components D, E, F, G, H in stream 11
9	Reactor temperature	42	MV for D feed flow (stream 2)
10	Purge rate (stream 9)	43	MV for E feed flow (stream 3)
11	Product separator temperature	44	MV for A feed flow (stream 1)
12	Product separator level	45	MV for total feed flow (stream 4)

13	Product separator pressure	47	MV for purge valve (stream 9)
14	Product separator underflow (stream 10)	48	MV for separator pot liquid flow (stream 10)
15	Stripper level	49	MV for stripper liquid prod flow (stream 11)
16	Stripper pressure	51	MV for reactor cooling water flow
17	Stripper underflow (stream 11)	52	MV for condenser cooling water flow

Table 3.2: Process faults in the TEP model. The IDV notation is used consistently with Downs and Vogel[1].

No.	Description	Type
IDV(1)	A/C feed ratio, B composition constant (stream 4)	Step
IDV(2)	B composition, A/C ratio constant (stream 4)	Step
IDV(3)	D feed temperature (stream 2)	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	Condenser cooling water inlet temperature	Step
IDV(6)	A feed loss (stream 1)	Step
IDV(7)	C header pressure loss-reduced availability (stream 4)	Step
IDV(8)	A, B and C composition (stream 4)	Random variation
IDV(9)	D feed temperature (stream 2)	Random variation
IDV(10)	C feed temperature (stream 4)	Random variation
IDV(11)	Reactor cooling water inlet temperature	Random variation
IDV(12)	Condenser cooling water inlet temperature	Random variation
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking
IDV(16)	Unknown	-

IDV(17)	Unknown	-
IDV(18)	Unknown	-
IDV(19)	Unknown	-
IDV(20)	Unknown	-
IDV(21)	The valve for stream 4	Constant position
IDV(22)	E feed temperature (stream 3)	Random variation
IDV(23)	A feed pressure (stream 1)	Random variation
IDV(24)	D feed pressure (stream 2)	Random variation
IDV(25)	E feed pressure (stream 3)	Random variation
IDV(26)	A and C feed pressure (stream 4)	Random variation
IDV(27)	pressure fluctuation in reactor CW re-circulating unit	Random variation
IDV(28)	pressure fluctuation in condenser CW re-circulating unit	Random variation

When using the MATLAB version of TEP, the fault initiation time and duration of simulation can be user-defined, along with the type of fault to be applied. In this study, all of the faults were set to be introduced at the 1000^{th} data point, and the 21 simulations were run until the 7200^{th} data point to observe the change in process conditions.

3.3.2 Glasso-MRF monitoring on TEP

Prior to the actual monitoring process, the iterative graphical lasso algorithm is implemented on the 50 process variables of the TEP to divide them into relevant groups. The resulting groups are visualized in Figure 2.3. The results and the variable description for each of the groups are the same as shown in Section 2.5.

After the separate groups subject to monitoring are defined, appropriately trained KDE models for each of the groups, as well as their monitoring limits have to be obtained. KDE for multiple variables usually suffer from the "curse of dimensionality", and a large number of data points are required to obtain a model with low error rates. To obtain the required number of data points for each of the monitored groups, the information given in Gonzalez et al.[2] is used. Upon fitting the data to an exponential equation and evaluating the model for 11 variables, approximately 70,000 data points are required to reach sufficient confidence levels. Thus the simulation under normal operating conditions is run until the $70{,}000^{th}$ data point, then a KDE model was trained for each of the groups using this dataset. The data from Gonzalez et al.[2], and the required number of data points calculated from the fitted equation is given in Table 3.3.

Table 3.3: Number of data points required for training multivariate KDE models. Values up to 6 dimensions were taken from Gonzalez et al.[2].

Dimension	Number of data points
1	40
2	84
3	175
4	366
5	765
6	1,600
11	67,890

Due to the relatively high dimensionality of each of the groups, and the large number of data points, the KDE training process takes some time. However, since this process is required only once for training the KDE models for each of the MRFs before the actual monitoring starts, it does not alter the real-time monitoring performance or the speed of the proposed method. The trained KDE models are used to calculate the monitoring limit. As mentioned in Rato and Reis[4], it is important to select the value of the significance level α so that all of the monitoring groups of subject to monitoring attain the same error rate in normal conditions. Thus the value of α was set so that each group showed an error rate of 5% in normal operating conditions.

The Glasso-MRF monitoring result for fault 1 is given in Figure 3.3, as an representative example. The monitoring results for each of the five groups are shown, where the probability density values calculated from the pre-trained KDE model are given as the black data points, and the monitoring limit for each group is given in red. Fault 1 makes a step change of the A/C ratio in stream 4 upon initiation at the 1000^{th} sample, altering the process variables downstream. The deviation of the process variables from normal operating values are evident in fault 1, and it can be seen that fault 1 is clearly detected in all of the five groups. All of the data points before the 1000^{th} sample are above the monitoring limit for all five groups, and then a large drop occurs so that the probability density values are below the monitoring limit after the 1000^{th} data point. It is notable that some samples in group 5 appear to have false negativity after the fault occurs, while showing a FDR of 93.35%. This is because the variables in group 5 are consisted of

quickly reacting variables in terms of control, such as flow streams, unit levels, and unit temperatures. It can be analyzed that the flow, level, and temperature controllers try to recover its normal operating points constantly after the fault occurs. While the FDR for group 5 maybe slightly lower than other groups, this does not cause a problem for the overall monitoring process since the monitoring results of other groups evaluate the status of the entire process as in a faulty condition.

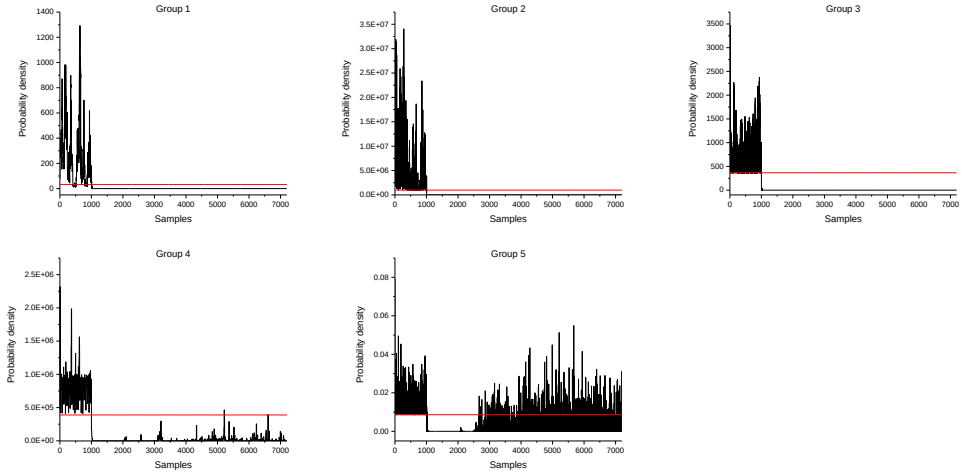
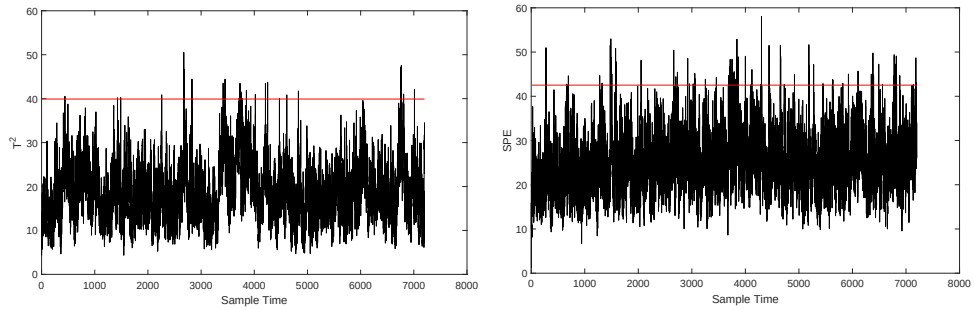


Figure 3.3: Glasso-MRF monitoring results for fault 1

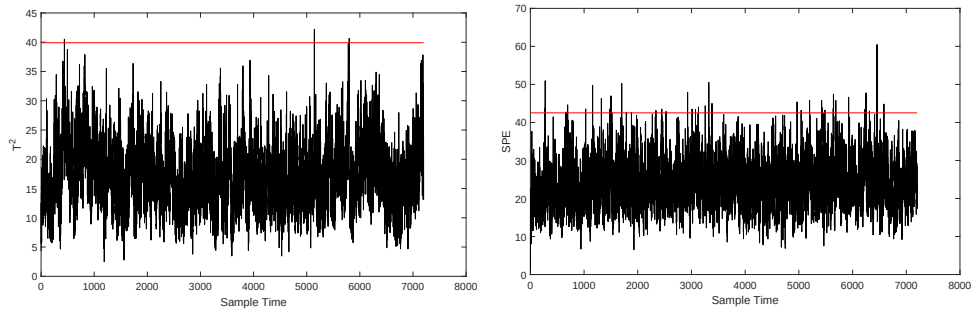
To compare the performance of Glasso-MRF monitoring to other monitoring methods, faults 9 and 15 were chosen for analysis. In Figures 3.4 and 3.5 the PCA monitoring results for the two faults are given, and in Figure 3.6 and 3.7, the Glasso-MRF monitoring results for the two faults are given. The conventional statistics, T^2 and SPE values, were used for PCA monitoring, and the number of PCs were set to 17, following the work by Rato and Reis[4]. Similar to Figure 3.3, the black lines represent the calculated monitoring statistic values for the monitoring method implemented, and the red lines represent the monitoring limits. It should be noted that due to the difference in monitoring methods, fault detecting conditions of PCA are opposite to that of MRF monitoring conditions, where a faulty condition occurs when the monitoring statistics are above the monitoring limit.



(a) T^2 for fault 9

(b) SPE for fault 9

Figure 3.4: PCA monitoring results for fault 9.



(a) T^2 for fault 15

(b) SPE for fault 15

Figure 3.5: PCA monitoring results for fault 15.

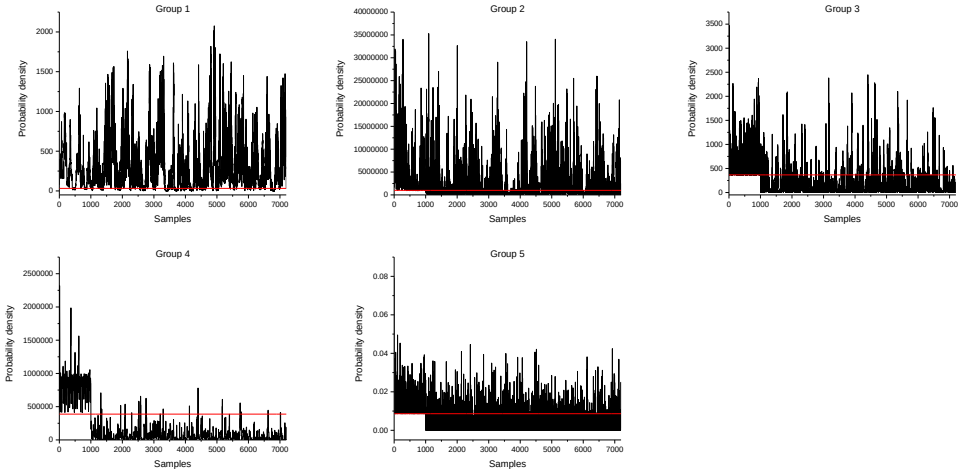


Figure 3.6: Glasso-MRF monitoring results for fault 9.

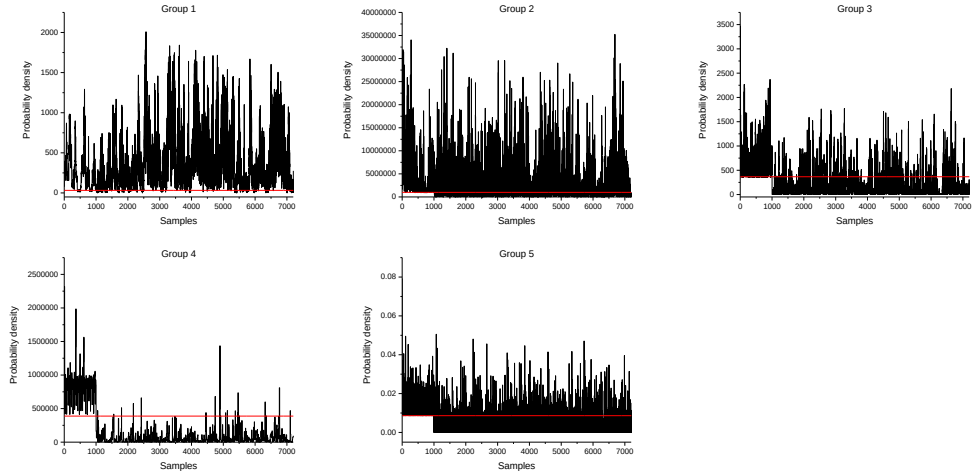


Figure 3.7: Glasso-MRF monitoring results for fault 15.

Fault 9 occurs as a random variation of the D feed (stream 2) temperature. As shown in Figure 3.4a and Figure 3.4b, PCA fails in detecting the fault, where most of the monitoring statistic values stay below the monitoring limit. Since there is no change in the data trend for the PCA monitoring statstics, it can be assumed that PCA fails at detecting the fault. In Figure 3.6, monitoring results for separate groups are shown for Glasso-MRF monitoring. While groups 1 and 2 seem to fail in effectively detecting the fault, with many false negativity, groups 3, 4, and 5 succeed in detection. Especially in group 4, the fault is detected with high accuracy, with a detection rate of 97.47%. This shows the advantage and effectiveness of Glasso-MRF monitoring, where by dividing the monitored variables into groups of similar characteristics or intercorrelation in terms of control or distance, different groups become separately sensitive to a specific fault. Whereas some groups might not react to a fault due to low correlation, groups containing variables related to the fault of interest show great performance in fault detection, increasing the overall FDA of the process.

Fault 15 occurs as valve sticking of the condenser cooling water valve. Similar to the case for fault 9, PCA monitoring fails in detecting the occured fault, showing no data trend change after the fault occurs, as shown in Figure 3.5a and Figure 3.5b. The results of the Glasso-MRF monitoring shows good fault detection performance as shown in Figure 3.7. Where the groups 1 and 2 are insensitive to the fault, but groups 3, 4, and 5 effectively detect the fault. The common features of fault 9 and 15 are that both faults are related to temperature maintenance of the process. Along with the fact that

group 4 mainly consists of composition values of the input and output flow streams, it can be assumed that temperature related variations result in only small changes of the monitored variables, making it difficult for the conventional process monitoring methods to detect the change. This is in accordance with the fact that temperature changes are quickly controlled, and directly results in composition disturbance in the TEP, since it affects the reaction rate. The FDA and FDR of all five groups, for each of the faults are presented in Table 3.4. The overall FDA and FDR are values from the group showing the best monitoring performance. Since error detection in one group qualifies as a faulty condition, this value can be deemed as the monitoring performance of the Glasso-MRF monitoring. It can be seen that, although individual group performance varies according to the type of fault and the variables related to them, Glasso-MRF monitoring shows stable monitoring results, showing over 95% FDA and over 96% FDR for all of the faults.

The Glasso-MRF fault detection results for all of the faults other than IDV(1), IDV(9) and IDV(15), are shown in Figures 3.8 through 3.32. It is notable that among the five groups, group 4 shows consistent fault detection results, while other groups fail in detecting some of the complex faults. This is due to the fact that group 4 is comprised of composition variables, and thus reacts to any form of process fault.

Table 3.4: FDA and FDR of the five groups for all 28 faults (FDA/FDR).

	Groups				
Fault No.	G1	G2	G3	G4	G5
IDV(1)	98.13/99.89	98.92/99.95	99.25/1.00	99.01/99.66	93.35/93.11
IDV(2)	97.92/99.65	98.71/99.71	99.19/99.94	99.29/99.98	98.97/99.65
IDV(3)	15.15/3.53	57.13/51.43	88.42/87.42	96.39/96.61	89.03/88.10
IDV(4)	19.80/8.93	98.94/99.98	88.82/87.89	96.61/96.87	89.53/88.68
IDV(5)	17.71/6.50	52.20/45.70	88.64/87.68	97.22/97.58	90.67/90.00
IDV(6)	98.21/99.98	98.93/99.97	99.24/99.98	99.15/99.82	99.26/99.98
IDV(7)	22.86/12.48	47.88/40.69	99.25/1.00	97.46/97.86	89.70/88.87
IDV(8)	96.35/97.82	98.13/99.03	99.07/99.79	99.29/99.98	98.56/99.16
IDV(9)	22.86/12.48	61.03/55.96	89.08/88.20	97.47/97.87	89.89/89.10
IDV(10)	25.04/15.01	97.15/97.90	89.88/89.11	97.72/98.16	91.10/90.50
IDV(11)	74.89/72.91	98.46/99.42	90.52/89.86	96.74/97.02	98.08/98.61
IDV(12)	50.99/45.15	88.20/87.50	89.86/89.10	96.28/96.48	91.77/91.28
IDV(13)	97.74/99.44	98.49/99.45	99.01/99.73	99.29/99.98	99.22/99.94
IDV(14)	23.59/13.34	98.93/99.97	88.99/88.08	97.17/97.52	98.86/99.52
IDV(15)	18.04/6.89	45.88/38.36	86.88/85.63	97.40/97.79	90.04/89.28
IDV(16)	15.23/3.63	42.69/34.66	88.79/87.86	96.53/96.77	89.20/88.29
IDV(17)	59.37/54.88	97.92/98.79	88.63/87.66	97.01/97.34	97.44/97.87

IDV(18)	28.47/19.00	95.92/96.47	89.56/88.74	97.68/98.11	91.49/90.95
IDV(19)	58.13/53.44	98.79/99.81	92.06/91.65	97.06/97.39	95.07/95.11
IDV(20)	96.58/98.10	97.50/98.31	98.33/98.94	99.01/99.66	98.76/99.40
IDV(21)	20.68/9.95	45.48/37.90	88.02/86.95	96.89/97.19	89.71/88.89
IDV(22)	22.09/11.59	63.41/58.72	87.57/86.44	96.63/96.89	90.07/89.31
IDV(23)	38.87/31.08	42.15/34.03	87.70/86.58	97.14/97.48	89.02/97.48
IDV(24)	97.00/98.58	95.43/95.90	90.65/90.02	97.89/98.36	96.17/98.58
IDV(25)	94.76/95.98	80.53/78.60	92.46/92.11	97.96/98.44	91.11/98.44
IDV(26)	54.38/49.09	63.10/58.36	98.88/99.56	96.88/97.18	89.78/99.56
IDV(27)	37.22/29.16	96.35/96.97	89.45/88.61	96.69/96.97	96.93/97.27
IDV(28)	19.55/8.64	58.46/52.98	89.47/88.65	97.28/97.65	89.82/97.65

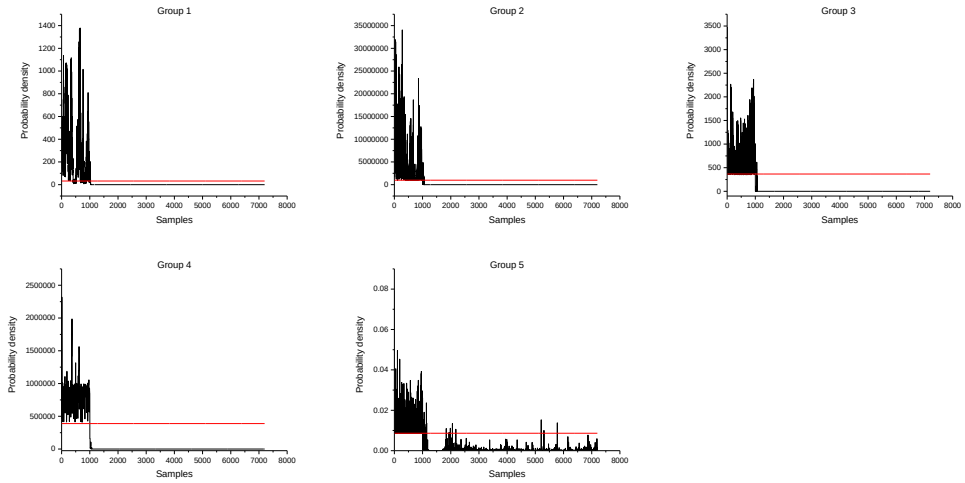


Figure 3.8: Glasso-MRF monitoring results for fault 2.

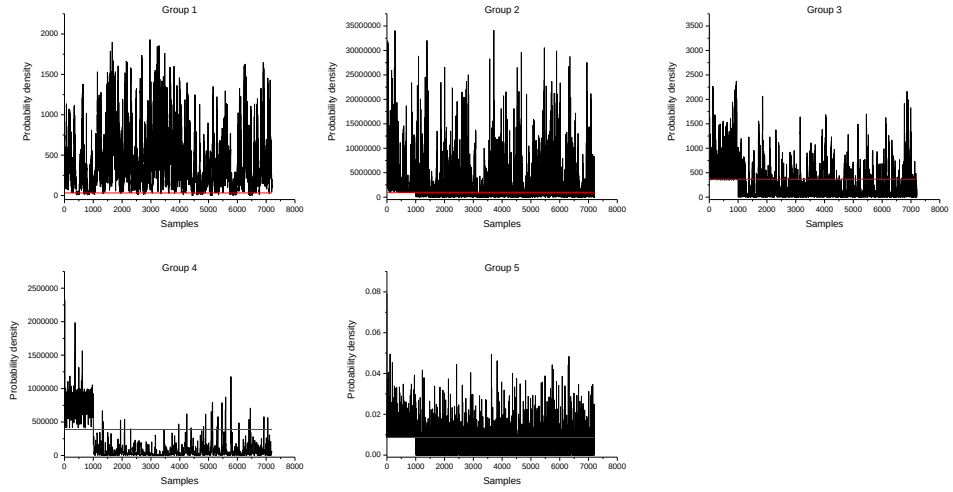


Figure 3.9: Glasso-MRF monitoring results for fault 3.

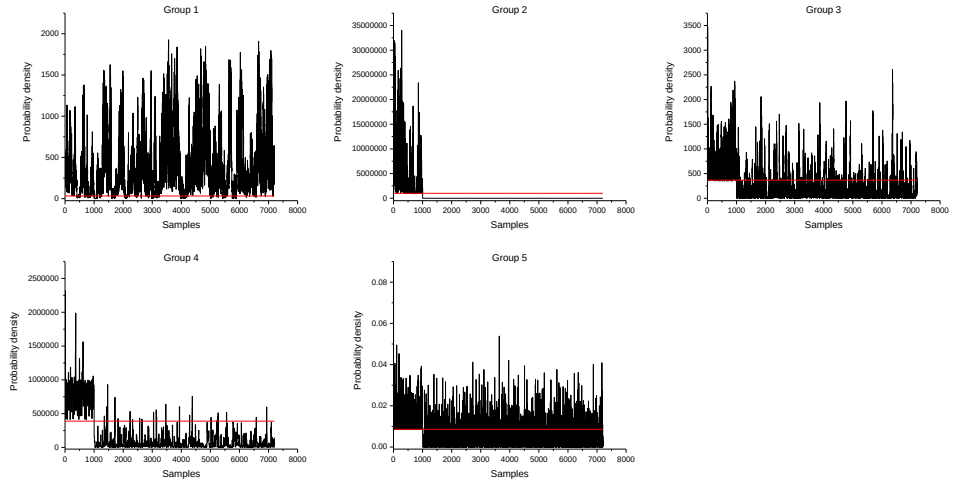


Figure 3.10: Glasso-MRF monitoring results for fault 4.

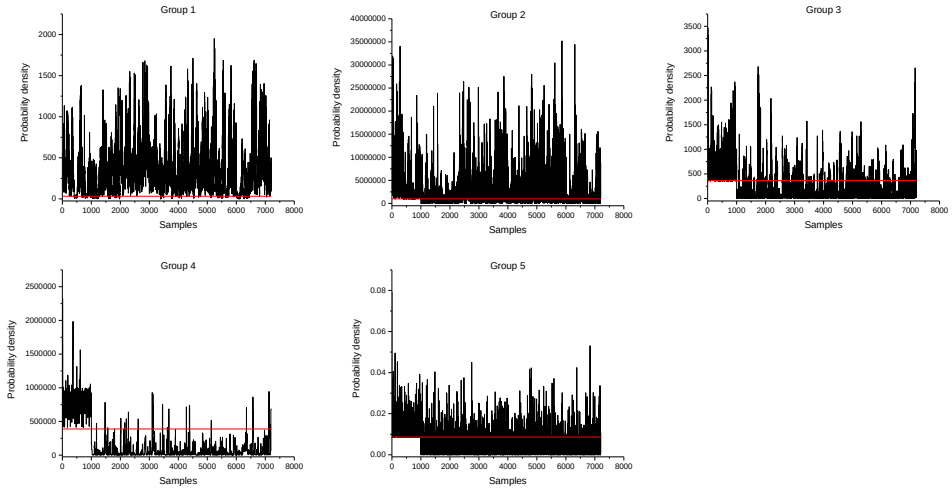


Figure 3.11: Glasso-MRF monitoring results for fault 5.

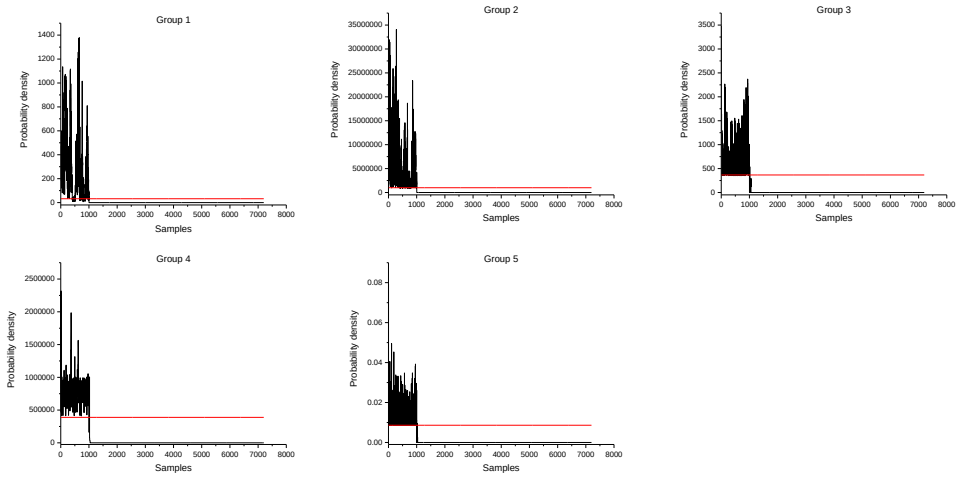


Figure 3.12: Glasso-MRF monitoring results for fault 6.

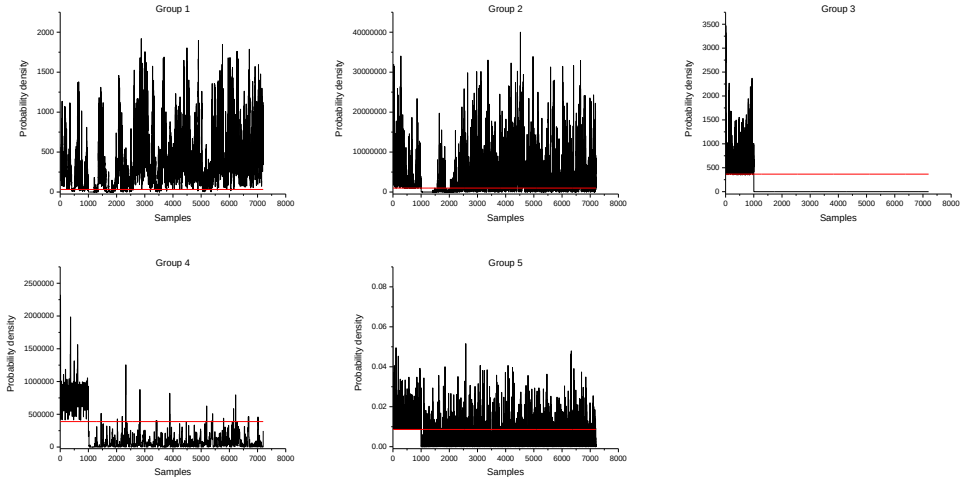


Figure 3.13: Glasso-MRF monitoring results for fault 7.

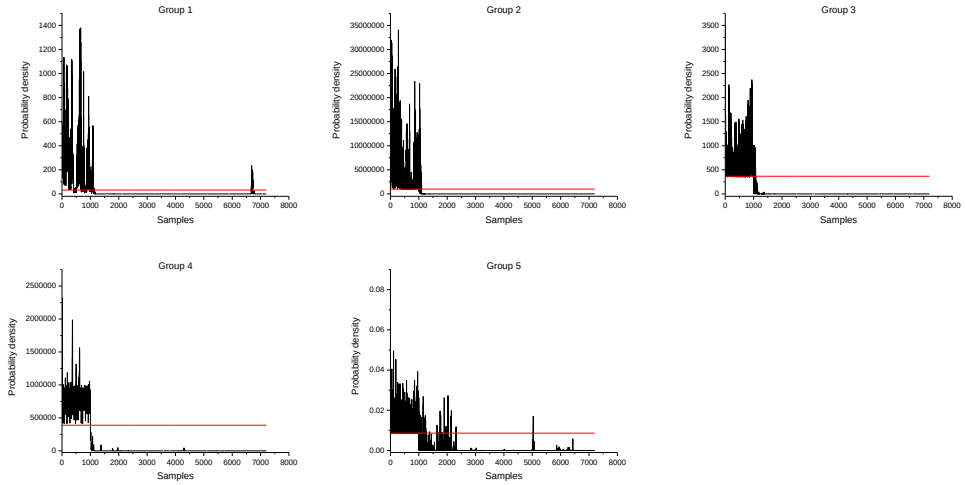


Figure 3.14: Glasso-MRF monitoring results for fault 8.

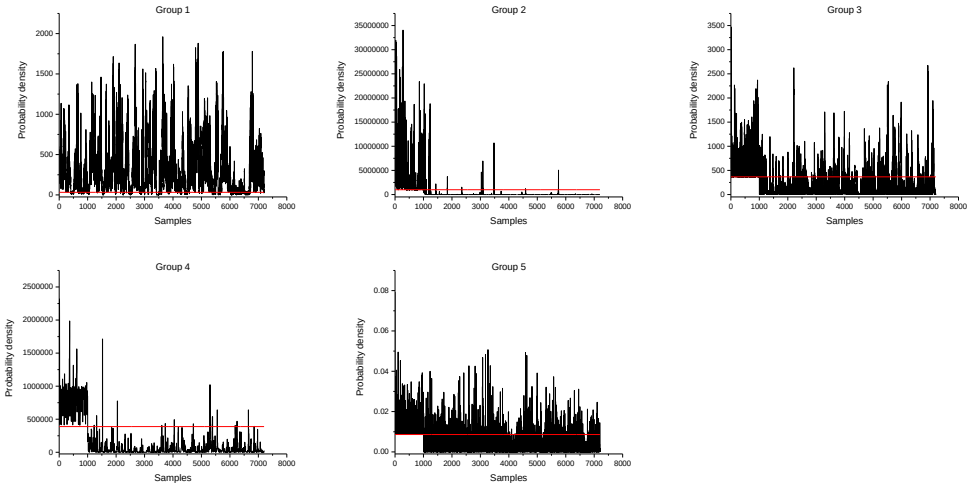


Figure 3.15: Glasso-MRF monitoring results for fault 10.

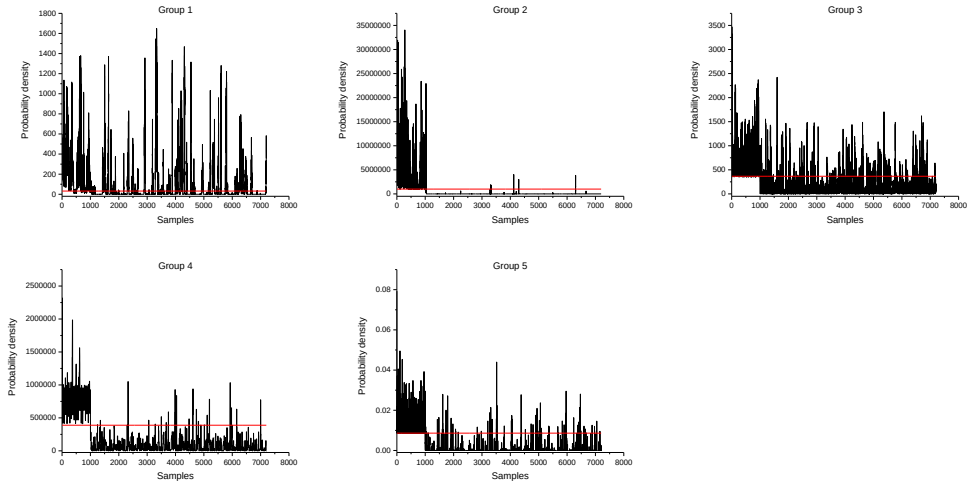


Figure 3.16: Glasso-MRF monitoring results for fault 11.

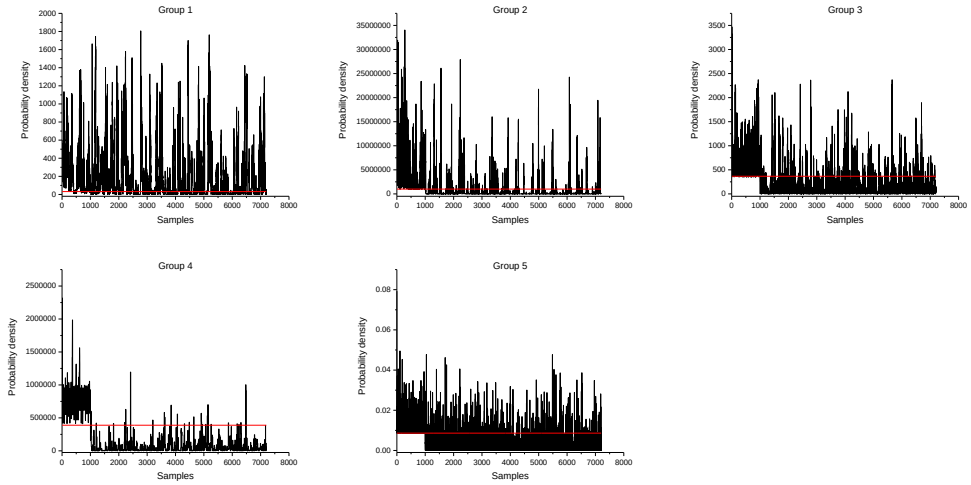


Figure 3.17: Glasso-MRF monitoring results for fault 12.

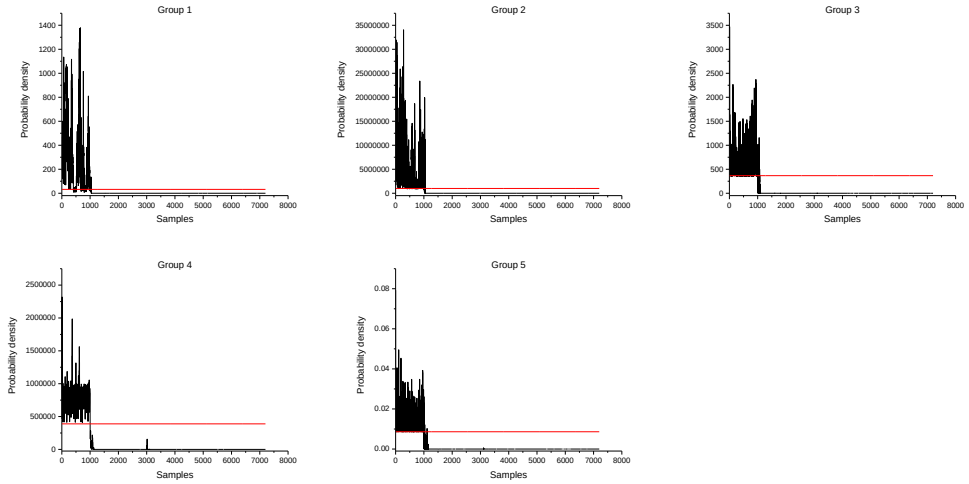


Figure 3.18: Glasso-MRF monitoring results for fault 13.

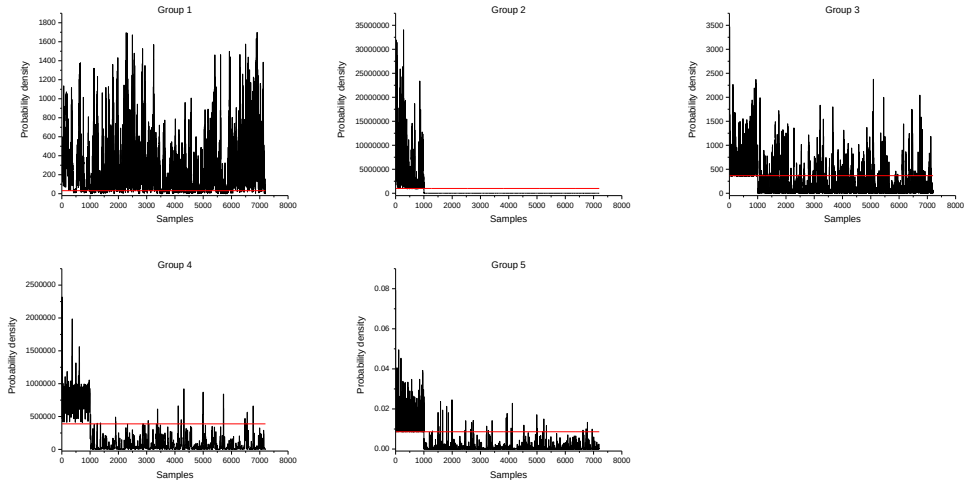


Figure 3.19: Glasso-MRF monitoring results for fault 14.

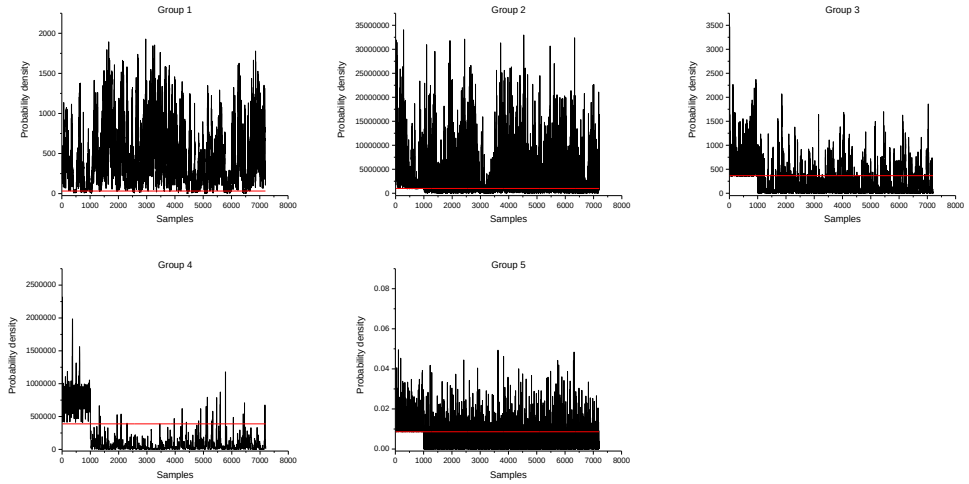


Figure 3.20: Glasso-MRF monitoring results for fault 16.

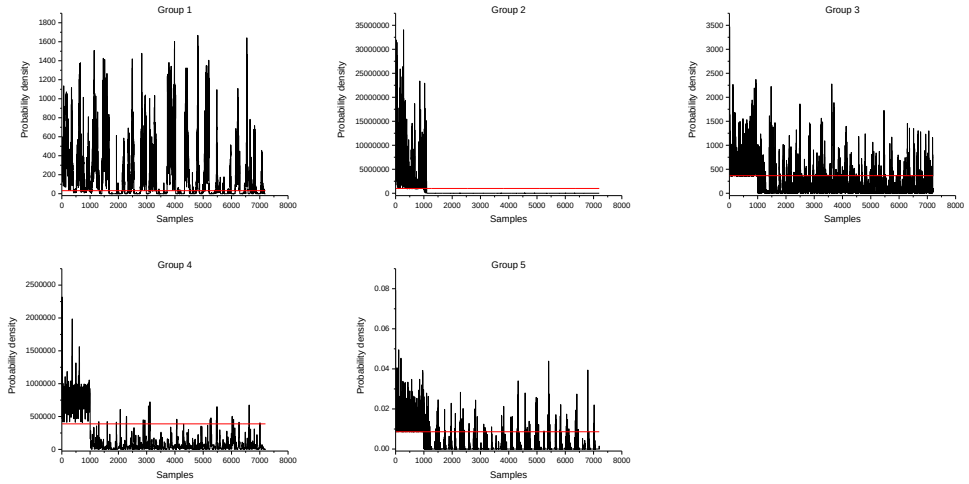


Figure 3.21: Glasso-MRF monitoring results for fault 17.

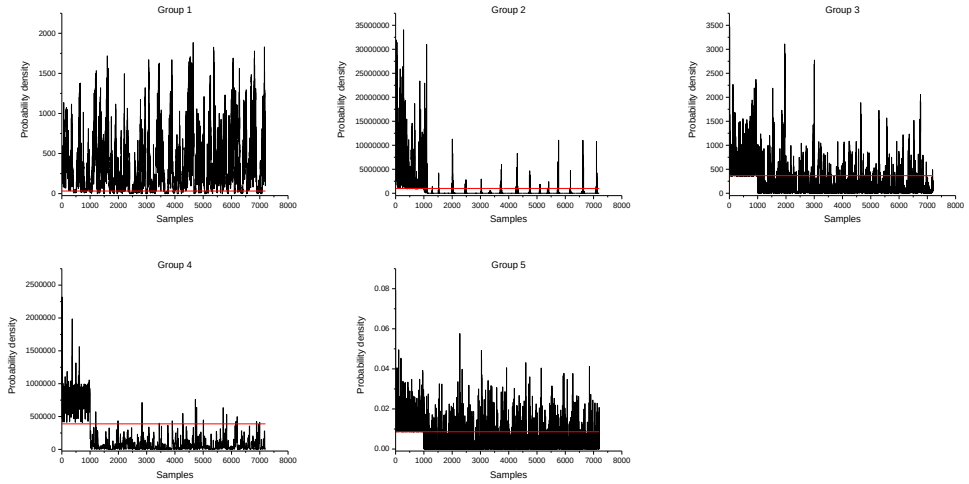


Figure 3.22: Glasso-MRF monitoring results for fault 18.

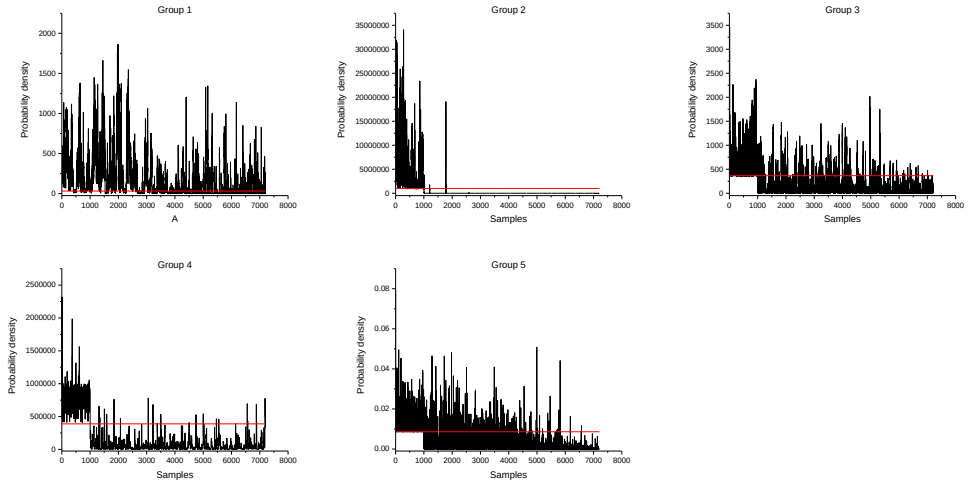


Figure 3.23: Glasso-MRF monitoring results for fault 19.

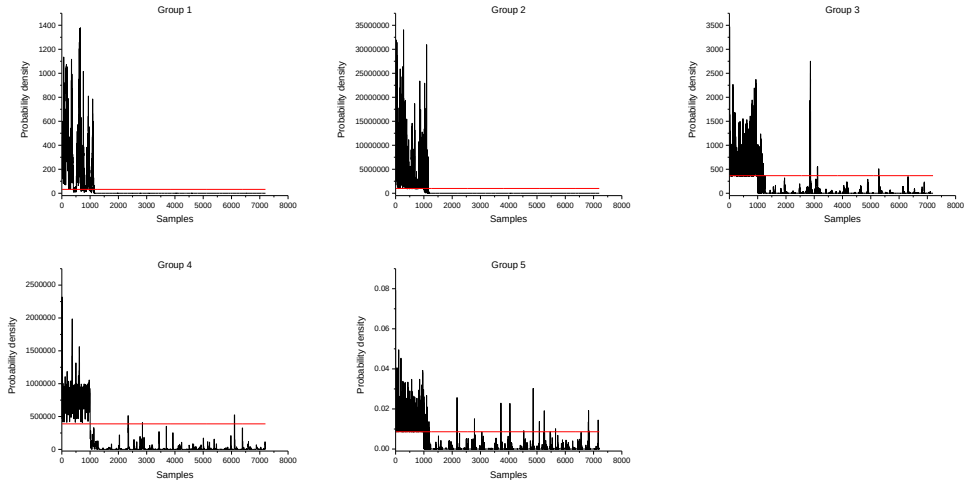


Figure 3.24: Glasso-MRF monitoring results for fault 20.

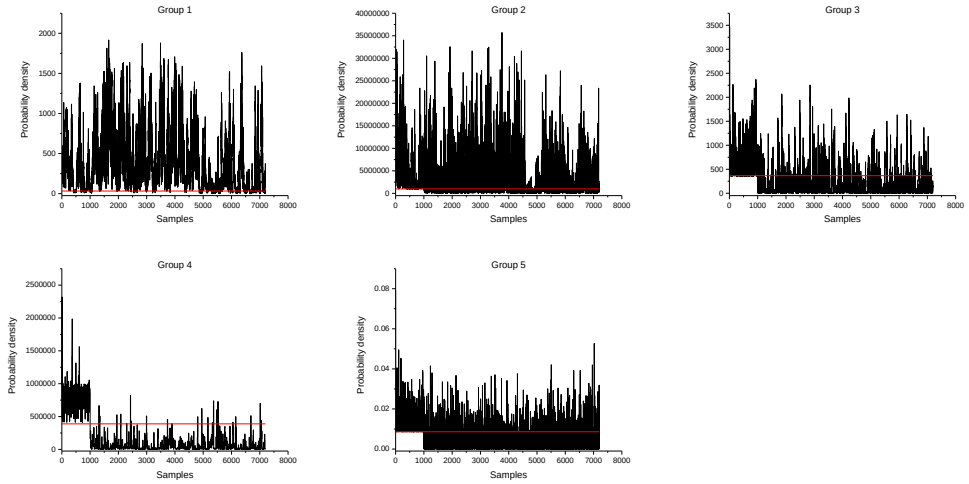


Figure 3.25: Glasso-MRF monitoring results for fault 21.

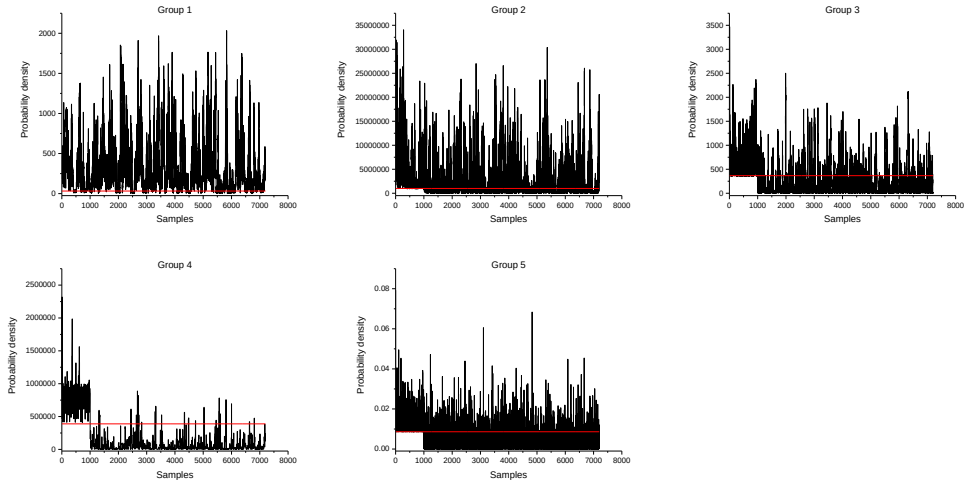


Figure 3.26: Glasso-MRF monitoring results for fault 22.

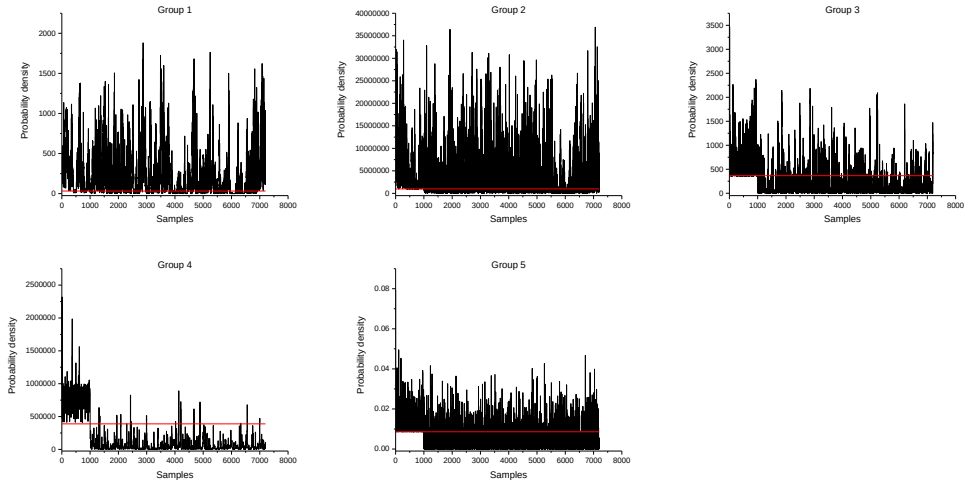


Figure 3.27: Glasso-MRF monitoring results for fault 23.

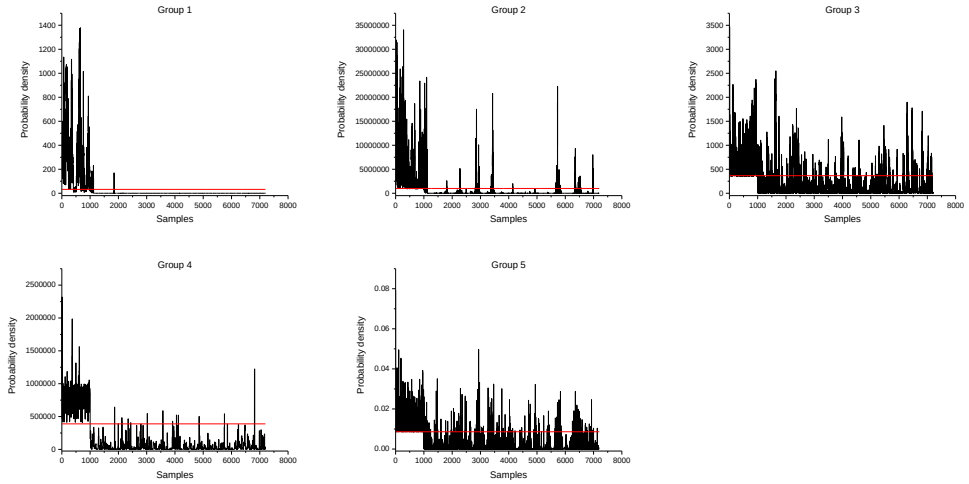


Figure 3.28: Glasso-MRF monitoring results for fault 24.

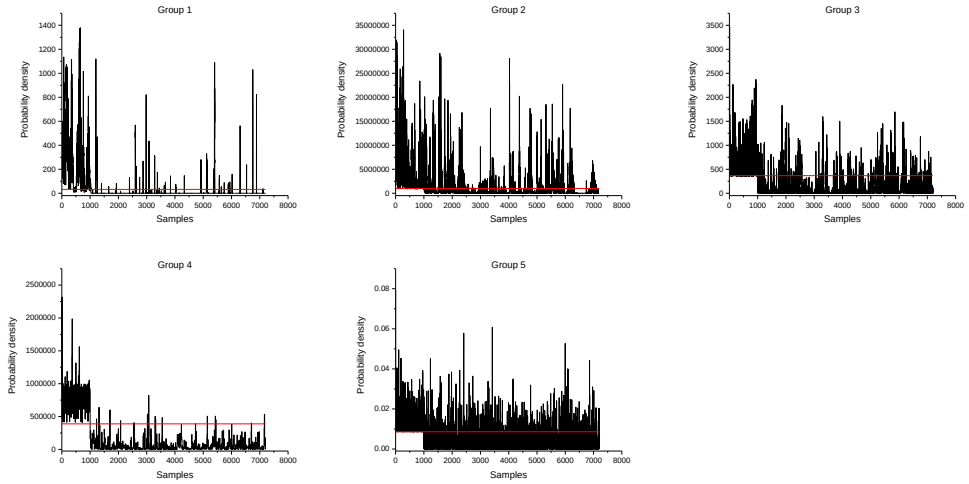


Figure 3.29: Glasso-MRF monitoring results for fault 25.

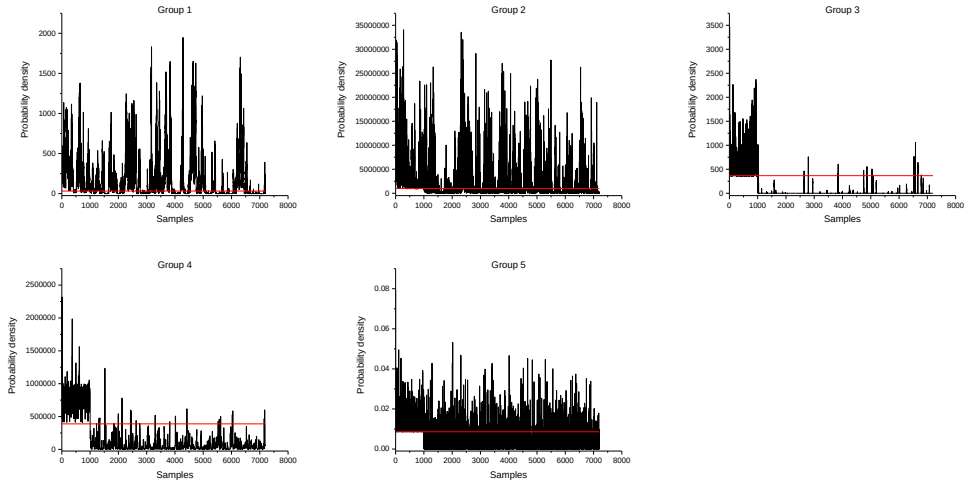


Figure 3.30: Glasso-MRF monitoring results for fault 26.

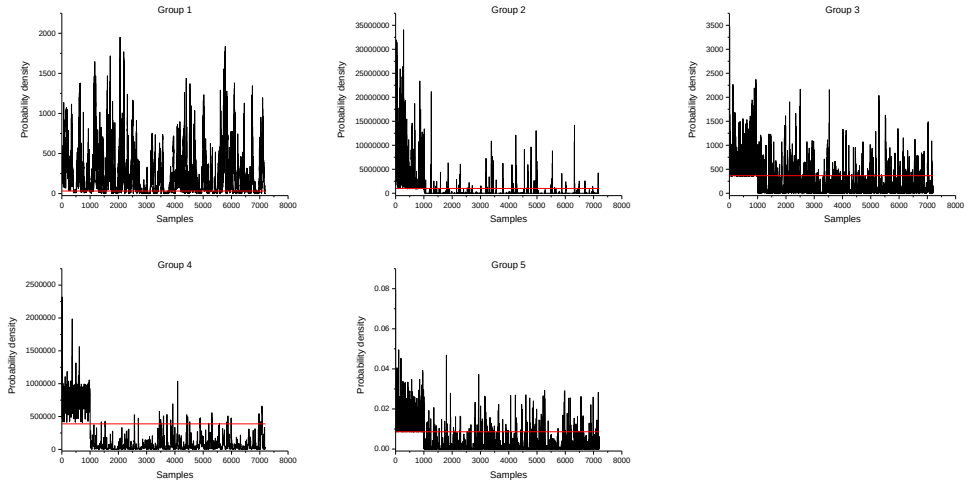


Figure 3.31: Glasso-MRF monitoring results for fault 27.

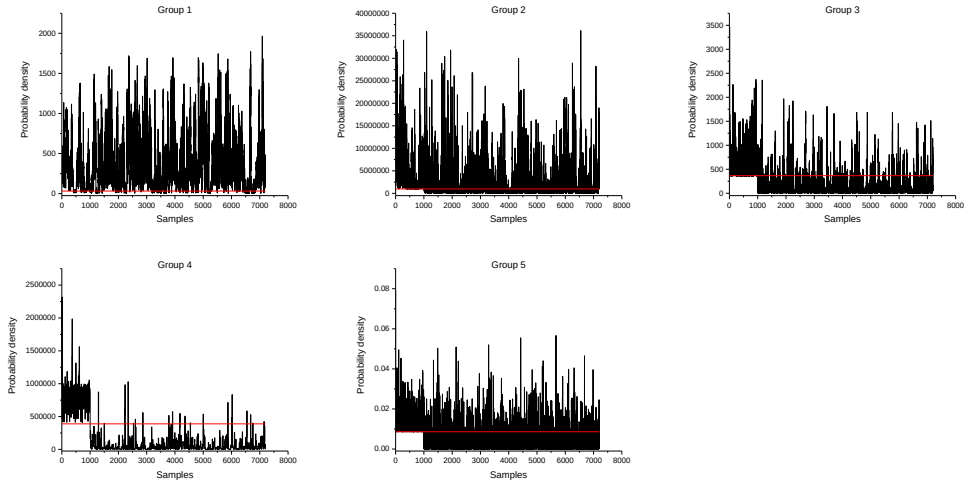


Figure 3.32: Glasso-MRF monitoring results for fault 28.

3.3.3 Fault detection accuracy comparison with other monitoring techniques

For the purpose of performance comparison, monitoring results for Glasso-MRF monitoring are shown along with other conventional and state-of-the-art monitoring techniques, for the first 21 programmed faults, in Table 3.5. Variant autoencoder based monitoring and the DPCA-DR method are chosen as representatives of state-of-the-art methods, since they show improved fault detection accuracy regarding IDV(3), IDV(9), and IDV(15), compared to conventional methods [5]. The FDA values of various methods are taken from Yan et al.[3] and the FDR values for the DPCA-DR method are taken from Rato and Reis[4].

Table 3.5: FDA (%) of PCA, KPCA, CAE[3], and Glasso-MRF monitoring, and the FDR (%) for DPCA-DR method[4] for the first 21 faults within the TEP model

	PCA		KPCA		CAE		DPCA-DR		MRF
Fault No.	T^2	SPE	T^2	SPE	T^2	SPE	T^2_{PREV}	T^2_{RES}	
IDV(1)	99.0	99.2	99.3	100	99.0	96.8	99.6	99.8	99.3
IDV(2)	95.7	98.3	98.3	99.3	98.8	96.4	98.5	98.3	99.3
IDV(3)	2.0	1.6	2.7	6.1	83.3	45.1	2.1	1.6	96.4
IDV(4)	0.2	97.4	30.0	100	49.2	98.3	99.8	99.9	98.9
IDV(5)	23.3	43.5	24.2	27.8	44.7	96.9	99.9	99.9	97.2
IDV(6)	98.7	100	100	99.4	99.5	100	99.9	99.9	99.3
IDV(7)	98.0	98.7	100	100	98.3	100	99.9	99.9	99.3
IDV(8)	94.2	97.5	97.9	97.3	97.8	96.6	98.5	98.1	99.3
IDV(9)	1.0	1.2	2.8	5.1	18.2	30.4	2.0	1.0	97.5
IDV(10)	4.5	72.0	45.0	45.3	65.3	89.4	95.6	93.3	97.7
IDV(11)	19.5	73.3	73.6	47.9	83.3	87.6	96.5	86.5	98.5
IDV(12)	92.1	98.1	99.0	98.5	99.8	99.4	99.8	99.8	96.3
IDV(13)	93.6	96.0	94.8	94.3	95.9	98.6	95.8	95.6	99.3
IDV(14)	89.8	98.8	100	99.5	100	97.1	99.8	99.9	98.9
IDV(15)	1.0	21.1	5.2	8.4	32.1	43.8	38.5	4.7	97.4
IDV(16)	1.2	65.8	42.0	36.4	47.9	85.8	97.6	94.5	96.5

IDV(17)	63.1	97.5	83.6	77.5	100	94.6	97.6	97.5	97.9
IDV(18)	88.2	92.1	89.9	90.5	100	97.6	90.5	90.0	97.7
IDV(19)	0.5	35.2	4.8	18.6	21.3	83.3	97.1	84.3	98.8
IDV(20)	13.2	64.5	51.5	57.6	99.9	98.6	90.8	91.6	99.0
IDV(21)	19.7	65.3	28.1	40.6	88.9	92.0	53.9	57.7	96.9

It is clear from the results in Table 3.5 that Glasso-MRF monitoring greatly outperforms all of the other monitoring algorithms, showing consistently high FDA for all of the faults. FDR values of Glasso-MRF monitoring are not shown in Table 3.5, but it can be seen from the results in Table 3.4 that Glasso-MRF monitoring outperforms the DPCA-DR method as well. For certain fault cases, such as fault 12, 17, and 18, FDA of Glasso-MRF monitoring is slightly lower than the CAE monitoring method proposed by Yan et al.[3], but the difference is negligible. The effectiveness of the proposed method is emphasized in faults number 3, 9, 15, 16, and 19. FDA for these faults are under 10 percent when using PCA and kernel PCA (KPCA), meaning that fault detection is barely possible in these cases. FDA increases for CAE, but still retain a value below 90 percent. For faults 9 and 15 the best performing method only shows FDA lower than 50 percent. However, with the use of the Glasso-MRF monitoring method, FDA for all of the faults are over 95 percent, showing the consistency of the proposed method. This high FDA results from the fact that the nonlinear, non-Gaussian characteristics of the variables are accounted for by using kernel density estimation, and that there is no reduction in dimensionality during the monitoring process, allowing even small changes in variable values to be effectively detected. Also, the monitoring strategy of separately monitoring the variables in different groups, which enables the small variations to be amplified so that they can be effectively monitored, contributes to the high FDA.

The false positive rate of Glasso-MRF was analyzed to check its sensitivity to

process noise. The false positive rates of various methods, including Glasso-MRF, are shown in Table 3.6. Analyzing the results, it can be seen that the Glasso-MRF shows good performance in terms of false positive rate, showing the second lowest value other than ICA. This is an expected result, since when determining the monitoring limit for separate groups, the limit was set according to a false alarm rate of 5%.

Table 3.6: False positive rates (%) of various methods [5, 6].

Methods	Average FPR (%)
PCA	6.13
DPCA	10.13
ICA	2.75
FDA	6.38
PLS	10
CVA	6.6
SCVA	6.5
Glasso-MRF	5.78

To emphasize the superior performance of using the Glasso-MRF for fault detection, a performance index, γ, is introduced. The definition of γ is given in the following equation:

$$\gamma = \frac{FDAofothermethod}{GlassoMRFFDA} \tag{3.4}$$

According to the definition, Glasso-MRF will have a γ value of 1 for all of the 21 faults. For other fault detection methods, a γ value smaller than 1 would mean that the method of interest shows inferior performance compared to the Glasso-MRF, whereas a value larger than 1 would mean that it shows superior performance. A plot of the performance index γ for all of the 21 faults regarding the five fault detection methods compared in Table 3.5, is shown in Figure 3.33. The values of γ are plotted in descending order so as to emphasize the difference in performance. The γ values of the previously proposed fault detection methods, are clearly shown to be negatively deviating from the γ value of the Glasso-MRF.

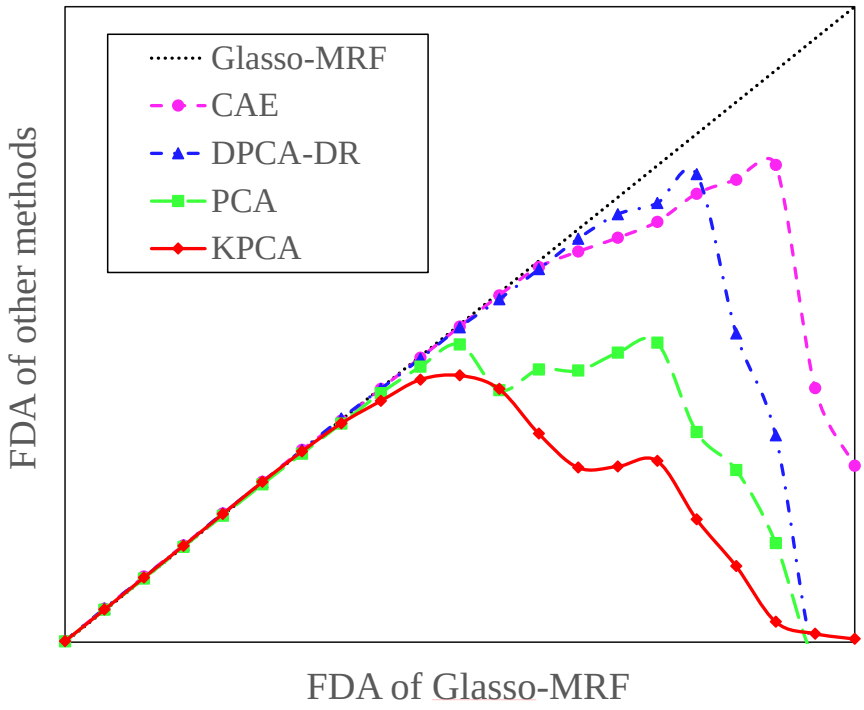


Figure 3.33: Performance index (γ) plot of the five fault detection methods.

3.3.4 Fault detection speed & fault propagation

While the fault detection performance of Glasso-MRF monitoring is evident, it also shows very fast fault detection speed, compared to conventional monitoring methods. The fault detection time for fault number 1, using PCA and Glasso-MRF monitoring, is shown in Table 3.7. The time difference in fault detection for the two monitoring methods is approximately 7 data points, which is equivalent to 252 seconds of monitoring time. This time difference in fault detection could result in a critical damage to the process in the actual industry, so it is important that the faults be detected as quickly as possible. Since Glasso-MRF monitoring dissects the entire set of variables into separate groups, allowing the individual variables to be analyzed without characteristics being ignored, fault detection is done in a more efficient manner. The time delay for all 28 faults of the TEP model was 1 sample point when applying the Glasso-MRF monitoring framework.

Table 3.7: Fault detection time of PCA and Glasso-MRF monitoring for fault 1.

Method		Detection time
PCA	T^2	1012
	SPE	1008
MRF	G1	1003
	G2	1003
	G3	1001
	G4	1002
	G5	1001

As another aspect of fault detection speed calculation, the fault propagation path can be analyzed using the fault detection speed of each of the groups, since the detection time for each group varies according to the magnitude of relevance of the composing variables in relation to the fault of interest. Fault cases number 3 and 15 are analyzed to illustrate the group-wise fault propagation path analysis performance of Glasso-MRF monitoring.

As shown in Table 3.8, the fault detection time of each group varies significantly for both of the fault cases. In fault number 3, where the D feed (stream 2) is changed step-wise, the detection time of the slowest group is delayed by 44 data points, compared to the group with the fastest detection time. It can be predicted that the fault propagates along the groups according to their fault detection time, allowing a more extensive understanding of the nature of the fault. The fastest groups, number 3 and 5, are composed of variables directly related to feed streams and reactor operating conditions, such as D feed (2), E feed (3), Recycle flow (5), Reactor level (8), and Reactor temperature (9). The number in parentheses represent the variable numbers from Table 3.1. The next group to detect the fault is group 4, which is composed mainly of feed composition variables from streams 6, 9, and 11. This can be seen as the change in feed flowrate of stream 2 affecting the compositions of purge and product streams, which is a natural propagation of the fault before the control structure starts working to mitigate the change. Group 2, which consists mostly of temperature variables and manipulated variables, starts detecting the fault at sampling point 1008. This can be

seen as the manipulated variables being activated to mitigate the change caused by the fault. Also, these manipulated variables alter the temperature values of the separator and stripper, altering the monitoring values of group 2 even more. Group 1 is consisted mostly of flow, and pressure variables, and the manipulated variables that alter the flow and pressure variables. It can be analyzed that, after the change in flowrate of stream 2 is altered, and has affected the flowrate, composition and temperatures of downstream units, the control structure for manipulating input flowrates are activated to minimize the change evoked by the change in stream 2. This analysis, provided by the detection time difference of the groups, is in accordance with the decentralized control structure described in Ricker[54].

For fault number 15, the detection time of the slowest group is delayed by 11 data points, compared to the group with the fastest detection time. This difference may seem small compared to the case of fault number 3, but it still evidently shows the fault propagation path. Fault 15 is related to the sticking of the condenser cooling water valve, leading to insufficient condensing of reactor outlets. This directly affects the product flowrates and compositions, resulting in immediate error detection in groups 3 and 5, followed by group 4. This is in accordance with the case of fault number 3, and is natural considering that the flowrate and composition are the first to be affected by flow phase changes. However, the order of fault detection of the final two groups are opposite to that of fault number 3, where group 1 detects the fault faster than group 2. This is due to the fact that group 1 is composed of pressure related variables, such

as Reactor pressure (7), Product separator pressure (13), Stripper pressure (16), and manipulated variable for purge valve (47). Sticking of cooling water valve leads to pressure increase, eventually making the alarm for group 1 to go off. Although the fault detection time of group 1 is slower than groups 3, 5, and 4, this is a natural result, since the control structure implemented[54] offers decentralized control of the variables, and flowrates are controlled to a setpoint whereas pressures are controlled to be within a certain range. As for group 2, it is the slowest to react since it mainly comprises of temperature related variables, reacting to the fault after the pressure related variables are affected. This different reaction of the variable groups compared to the case of fault 3, reflects the difference in the feature of the initiated fault.

As can be seen in the two examples, with a quick analysis of the difference in fault detection time of the groups and the variables consisting each group, the propagation process of the initiated fault, and the activation of control structures can be predicted. This allows the user to quickly analyze the specifics and prevent further propagation of the fault, leading to safer plant operation.

Table 3.8: Detection time and variable analysis of the five groups for fault 3 and 15.

Fault Case	Groups	Order of detection	Initial detection time (samples)
IDV(3)	G3	1	1001
	G5	1	1001
	G4	3	1002
	G2	4	1008
	G1	5	1045
IDV(15)	G3	1	1001
	G5	1	1001
	G4	3	1002
	G1	4	1003
	G2	5	1012

Chapter 4

Process Fault Diagnosis via Markov Random Fields Learning and Inference

4.1 Introduction

Process monitoring, or fault detection and diagnosis (FDD), is an essential part of product quality management and safe plant operation. FDI methods can be largely divided into two types, knowledge-based and data-based, but since it is difficult to know a priori the complex variable relationships in process systems, and because insignificant information may disturb the modelling procedure, data-based methods are preferred over knowledge-based methods[51]. There exist various methods for efficient process fault detection, from the conventional methods of using PCA to state-of-the-art methods using Markov random field (MRF) inference (Kim et al.[55]). These models are specialized in detecting the occurence of a fault, and show good performance when

applied to actual process plants. However, process fault diagnosis, namely the isolation of the root fault node and detection of the fault propagation path, is as important as the detection of the fault, but not many methods exist specialized for this purpose.

One of the most widely used methods for fault isolation is the use of projectory methods, which have been studied along with the development of process monitoring methods. Methods such as PCA [39–41, 56], independent component analysis (ICA) [43, 44], Fisher discriminant analysis [42] and partial least squares (PLS) citeCHIANG2000243 all focus on extracting the key features of the given data, and using these key components to calculate the contribution of each variable with respect to the occuring fault. The process variable with the largest contribution to the fault can be deemed as the root cause variable, and these sorts of techqniues are well-shown in studies such as whatever et al. However, these methods share the same limitations as that of projection-based monitoring. Since these projectory tecnhniques linearly decompose the variable dimensions, they are incapable of capturing nonlinear, non-Gaussian relationships between variables, which are common in process engineering data sets. Recent studies made efforst to resolve this issue, such as the work by Bao et al.[10] and Luo et al.[11], where they made use of sparse global-local preserving projections to model process variables. Their work succeeded in obtaining better representations of the process variables, resulting in better results with respect to process monitoring and fault isolation. However, these works are still limited in that they are not able to take into account the intricate relationships among all of the variables, which is

due to the characteristic of dimensionality reduction. While dimesionality reduction allows efficient monitoring variables, the individual characteristics of the variables are inevitably ignored. This results in performance degradation of fault isolation, especially in faults with minor amplitudes. Thus, more developed methods involving all of the variables are required.

Other methods making use of all the variable relationships have been proposed as well. These methods include the use of Granger causality, and transfer entropy. Granger causality is a widely used method for determining the dependence among varibles, based on vector autoregressive models (VAR). The causal relationship of two variables can be calculated according to the time-delayed influence of the the VARs representing the individual variables. Various studies have made use of the Granger causality in fault isolation[57, 58], but they are inheritantly limited by the fact that VARs are linearly regressed models of process data. Thus, they are not able to capture the nonlinear and non-Gaussian aspects of the variables. To overcome this limitation, other studies use transfer entropy, which is known as the nonlinear generalized form the Granger causality[59]. Transfer entropy is a term proposed by Schreiber[60]. Extending the mutual information term used in information theory, the conditional probabiltiy of calculating the entropy value of one node with respect to the other is calculated. Using this value, the direction of dependence can be determined, according to the direction having the higher entropy value. Since it is based on conditional probability calculation among variables, it is not limited by linear characterizations, and many

studies suggest methods for isolating faults using transfer entropy[61–65]. However, the main limitation of using transfer entropy is that the directional relationpships among variables are limited to bivariate pairs. Due to this characteristic, the complex relationships among process variables may not be effectively captured, and the fixation of direction of the variables may result in biased estimation of the direction of fault propagation. Also, since the computation of transfer entropy requires numerous number of samples, the computational load is large and thus can only be calculated after the occurence of a fault, limiting the use of the technique for post-analysis, not real-time analysis.

Recently, few studies made use of probabilistic graphical models (PGM) for efficent fault isolation. Use of PGMs allows intuitive modelling of the monitoring variables, as well as take into account the complex relationships among variable sets, not just between variable pairs. Verron et al.[66] proposed a FDI method based on Bayesian network fault detection and causal decomposition of the T^2 statistic. Decomposition of T^2 allows the causal relationships inferred by the Bayesian network to analyze the root cause variable of the occuring fault. Mori et al.[14] proposed a PGM identification methodology for root-cause diagnosis in industrial processes. This method learns the structure of the Bayesian network with respect to process data, by means of maximizing Bayesian scores for alternative causal networks. Then, using the abnormal likelihood index and the Bayesian contribution index, the causality structure is able to detect the fault as well as determine the root cause variable, and also allows fault propagation

path analysis. While these methods provide worthy perspective into the use of PGMs in fault isolation, they are limited in the use of Bayesian networks, which can only express limited forms of process networks. Also, the study by Verron et al. [66] only calculates a single network structure, which limits the adaptability of the model to various types of faults, and the study by Mori et al. [14] only calculates the contribution index for a single run of the fault, where the performance can be greatly limited by sample range selections.

In this chapter, a novel methodology using MRF modelling and inference is proposed for fault diagnosis in process systems, including fault isolation, or root fault variable detection, and the propagation path analysis of faults [9]. First the monitored variables are used to construct a fully-connected MRF, then the graphical lasso algorithm is implemented to obtain a sparse structure of the MRF. Afterwards, using a recently published learning and inference algorithm for MRFs consisting of continuous variables, the kernel belief propagation (KBP) [67, 68], the conditional marginal probability of the MRF is learned within the reproducing kernel Hilbert space (RKHS). For each sample, the conditional marginal probability values are converted into contribution values, to analyze the contribution of each variable. Using the contribution value the dominantly effective variable is deemed as the root variable, and as the samples go on, the order of contribution changes, showing the propagation of the occuring fault.

The following sections are organized as follows. In Section 2, preliminary contents, such as formulation of the MRF and the kernel belief propagation algorithm, are

introduced. In Section 3, The proposed methodology is introduced, which makes use of the MRF modeling, graphical lasso, and the KBP algorithm. Also, the newly defined contribution value induced from conditional marginal probability is introduced. In Section 4, the proposed method is applied to two processes, the two tank process and the Tennessee Eastman process (TEP), and the results of fault isolation and fault propagation analysis are shown. Then the paper is concluded in Section 5.

4.2 Preliminaries

4.2.1 Probabilistic graphical models & Markov random fields

The preliminary material regarding PGMs and MRFs are the same as shown in Section 2.4.

In a system where the variables are modelled as MRFs, the main interest would be to calculate the marginal probability of the node of interest, with respect to a specific observation. This process is called inference, and is only possible after learning the parameters of the MRF. Thus the main tasks when dealing with MRFs are the computation of the partition function Z, and parameter learning of the potentials consisting the MRF. The most widely used parameter learning algorithm for MRFs is the belief propagation algorithm, where the conditional probability of one node affecting another is modelled into a message, and then all of the messages from one node to another are iteratively calculated until convergence. However, most belief propagation algorithms are designed for inference in MRFs with discrete variables, and thus an

algorithm capable of computing the belief propagation process of continuous variables needs to be used when dealing with process systems.

4.2.2 Kernel belief propagation

The learning and inference of MRFs are computationally expensive compared to Bayesian networks, owing to the existence of the partition funciton. Various algorithms exist that allow efficient inference of MRFs, which include the junction tree algorithm, belief propagation algorithm, and the variational inference algorithms [15]. Since MRFs are mainly used in the modeling of images and gene networks, most of the learning and inference algorithms are intended for use on discrete variables.

A few studies exist which try to implement the belief propagation algorithm for continuous variables, such as the nonparametric belief propagation proposed by Sudderth et al.[69], which performs inference in Gaussian mixture models, the work by Wang et al. [70] where polynomial potentials are used for continuous MRF inference, and the particle belief propagation proposed by Ihler and McAllester[71], which creates a set of particles for each variable by sampling from the estimated posterior marginal, then using these particles to represent distributions of MRFs. However these methods are limited in the expression of diverse forms of nonlinearity other than the Gaussian, which are abundant in process variables. Also, these methods assume that the potentials are pre-specified by the user, meaning that they do not learn the model from training data. Thus they are only useful in systems having certain forms of continuous

variables [72, 73].

The most recently proposed method of inference for continuous MRFs is the kernel belief propagation (KBP), proposed by Song et al.[67, 68]. KBP is founded upon nonparametric representations for graphical models, where the marginals are expressed as Hilbert space embeddings and conditionals are expressed as embedding operators. Using this formulation the graphical model can be defined solely on the basis of the feature space representation of variables consisting the MRF, which allows the distributions on variables with complex structure and in high dimensions to be processed with ease. The main idea of KBP is to use the kernel trick with positive definite kernels, such as the radial basis function (RBF) kernel, to represent the messages of the belief propagation algorithm as functions in a reproducing kernel Hilbert space (RKHS). This allows the message update processes to become linear operations within the RKHS, without requiring any assumptions regarding the nonlinearity or parameterization of the MRF, except for the variables to be positive. The KBP algorithm consists of two parts: first the RKHS representations of the relations between variables are learned directly from training data, which removes the need for an explicit parameteric model. Secondly, the conditional marginal probabilities are computed through the belief propagation inference process, based on the learned relations.

Assume a MRF $G = (V, E)$ with nodes $V = 1, \cdots, n$, connected by edges in E, where each node $s \in V$ is associated with a random variable X_s on the domain X, and $\Gamma_s = \{t|(s,t) \in E\}$ is the set of neighbors of the node s with cardinality of $d_s = |\Gamma_s|$.

In a pairwise MRF, the joint distribution of the variables $X = X_1, \cdots, X_{|V|}$ is assumed to factorize according to a model as shown in Equation 4.1,

$$P(X) = \frac{1}{Z} \prod_{(s,t)\in E} \psi_{st}(X_s, X_t) \prod_{s\in V} \psi_s(X_s) \tag{4.1}$$

where $\psi_s(X_s)$ and $\psi_{st}(X_s, X_t)$ are the node and edges potentials, resepectively. Following the iterative procedure of belief propagation, the message m_{ts} passed from node t to node s is calculated based on messages m_{ut} from all neighboring nodes u of t except s, which can be shown as:

$$m_{ts}(X_s) = \int_{\chi} \psi_{st}(X_s, X_t)\psi_t(X_t) \sum_{u\setminus s} m_{ut}(X_t) dX_t \tag{4.2}$$

Thus the KBP algorithm iteratively calculates the messages until a fixed point, m^*_{ts} is obtained for all of the messages. Upon convergence, the belief at a specific node s can be computed as:

$$\mathbb{B}(X_s) = P^*_{X_s} \prod_{u\in\Gamma_s} m_{us}(X_s) \tag{4.3}$$

where the unconditional marginal $P^*_{X_s}$ can be computed using kernel density estimation.

The KBP represents the node and edge potentials as nonparametric functions learned from data, so that non-Gaussian statistical features can be captured. Since it is computationally infeasible to iteratively calculate Equation 4.2 to obtain the belief values, the strategy of KBP is to use the kernel trick to calculate the message update procedure within the RKHS.

The process of inducing the message update sequence as a linear operation in RKHS starts by defining the message at node S, given it is in the RKHS $\mathbb{F}$ of functions on the separable metric space $\mathbb{X}$:

$$m_{ts}(x_s) = \langle m_{ts}(\cdot), k(x_s, \cdot) \rangle_{\mathbb{F}} \tag{4.4}$$

where the kernel $k(x_s, x_{s'})$ is unique positive definite with the reproducing property $\langle m_{ts}(\cdot), k(x_s, \cdot) \rangle_{\mathbb{F}} = m_{ts}(x_s)$. If the Gaussian RBF kernel is used, the kernel is defined as $k(x_s, x_{s'}) := \exp(-\sigma \| x_s - x_{s'} \|^2)$. The advantage of this assumption is that the update procedure can be expressed as a linear operation in the RKHS, and results in new messages that are likewise RKHS functions. Then the message update from one external node, can be expressed using the reproducing property of the RKHS, as shown in Equation 4.5.

$$\begin{aligned} m_{ts}(x_s) &= \mathbb{E}_{X_t|x_s}[m_{ut}(X_t)] \\ &= \langle m_{ut}, \mathbb{E}_{X_t|x_s}[\phi(X_t)] \rangle_{\mathbb{F}} \end{aligned} \tag{4.5}$$

The second element of the inner product expressed in Equation 4.5, is the feature space embedding of the conditional distribution $\mathbb{P}(X_t|x_s)$, and is defined as $\mu_{X_t|x_s} := \mathbb{E}_{X_t|x_s}[\phi(X_t)]$. Since the direct computation of the feature space embedding allows the message updates to be obtained with simple inner product operations, Song et al. [68] proposes an expression for the conditional distribution embedding via the covariance

operator $C_{X_s X_t}$. Let $U_{X_t|X_s} := C_{X_s X_t} C_{X_s X_s}^{-1}$ such that for all $f \in \mathbb{F}$,

$$\begin{aligned} \mathbb{E}_{X_t|x_s}[f(X_t)] &= \langle f, \mathbb{E}_{X_t|x_s}[\phi(X_t)] \rangle_{\mathbb{F}} \\ &= \langle f, \mu_{X_t|x_s} \rangle_{\mathbb{F}} \\ &= \langle f, U_{X_t|X_s} \phi(x_s) \rangle_{\mathbb{F}} \end{aligned} \tag{4.6}$$

Using the expression in Equation 4.6, the message updates can be expressed as linear operations in feature space, as in Equation 4.7:

$$m_{ts}(x_s) = \langle m_{ut}, U_{X_t|X_s} \phi(x_s) \rangle_{\mathbb{F}} \tag{4.7}$$

Equation 4.7 shows the message update from a single variable, but typically in a MRF a node is surrounded by multiple neighboring nodes. Extending the single-node message update to multiple nodes, the definition of tensor product RKHS, $\mathbb{H} := \bigotimes^{d_t - 1} \mathbb{F}$, is used to express the product of incoming messages as a single inner product. The multiplication of multiple incoming messages using the tensor product RKHS can be expressed as shown in Equation 4.8:

$$\prod_{u \setminus s} m_{ut}(X_t) = \langle \bigotimes_{u \setminus s} m_{ut}, \xi(X_t) \rangle_{\mathbb{F}} \tag{4.8}$$

Then the message update from node t to node s can be expressed as:

$$\begin{aligned} m_{ts}(x_s) &= \langle \bigotimes_{u \setminus s} m_{ut}, \mathbb{E}_{X_t|x_s}[\xi(X_t)] \rangle_{\mathbb{H}} \\ &= \langle \bigotimes_{u \setminus s} m_{ut}, U_{X_t^{\otimes}|X_s} \phi(X_s) \rangle_{\mathbb{H}} \end{aligned} \tag{4.9}$$

Using the empirical estimate of the conditional embedding operator, $U_{X_t^{\otimes}|X_s} = \Phi^{\otimes}(L^T + \lambda m I)^{-1}\Upsilon^T$, where Φ and Υ are the feature matrices, $L = \Upsilon^T\Upsilon$, and λ is a regularization parameter for stablizing the calculation, the message update procedure can ultimately be expressed as:

$$\hat{m}_{ts}(x_s) = (\bigodot_{u\setminus s} K\beta_{ut})^T(L + \lambda m I)^{-1}\Upsilon^T\phi(x_s) \tag{4.10}$$

where $\odot$ is the elementwise vector product, K is the Gram matrix of each variable, $\phi(x_s)$ is the feature map, and β_{ut} is a factor for representing the messages as linear combinations of the training features, defined as $\hat{m}_{ut} = \Phi\beta_{ut}$. Using Equation 4.10, the message update process can be computed by calculating the Gram matrix K and $(L + \lambda m I)^{-1}$, then combining them together into a linear operation. Further details regarding the KBP algorithm can be found in Song et al.[67] and Song et al.[68].

The biggest strength of the KBP is that it can be used for MRF inference consisting of continuous variables, for any positive definite kernel available within the kernel Hilbert space with a reasonable computational cost of $O(m^2 d_{max})$, where m is the number of training samples and d_{max} is the maximum degree of a node in the graphical model. Thus this algorithm is implemented throughout the study as the main MRF learning and inference algorithm.

4.3 Fault Diagnosis via MRF Modeling

The proposed methodology makes use of MRF modelling and the KBP algorithm explained in the previous section to devise a strategy for calculating the contribution of each variable, with respect to the activated fault. First, the process variables are modelled into a fully-connected MRF, as the structure of the MRF is unknown. Then, to ease the computational burden of KBP, the structure of the MRF has to be learned. Structure learning of MRFs are usually infeasible, and thus MRFs are implemented in situations where the structure is known beforehand. In this study, the approximate structure of the process variable MRF is obtained using the graphical lasso. Graphical lasso assumes that the MRF of interest is a Gaussian graphical model, which may not always be the case for process variables. However, the purpose of using graphical lasso in this study is to cut off irrelevant variables and to eliminate edges pertaining weak relationships, so that the computational burden of KBP may be mitigated as much as possible. Thus the optimality of the obtained MRF structure is irrelevant, and so it is implemented as an approximation.

After determining its structure, the MRF is non-parametrically trained using the KBP algorithm with respect to process data in normal conditions. Then using the trained model, the conditional marginal probability of each of the variables for every sampled fault data are calculated, which considers the message updates provided by other variables. The iterative belief propagation calculation of the KBP algorithm is implemented until all of the messages sent from one variable to another are converged,

then the messages are multiplied to the marginal belief of individual variables to obtain the conditional marginal probability. Afterwards these marginal probabilities are processed into contribution values, which are used for analyzing the root cause variable and the fault propagation path.

The entire process of the methodology is shown as a flowchart in Figure 4.1. Each step of the method is explained in detail in the following subsections.

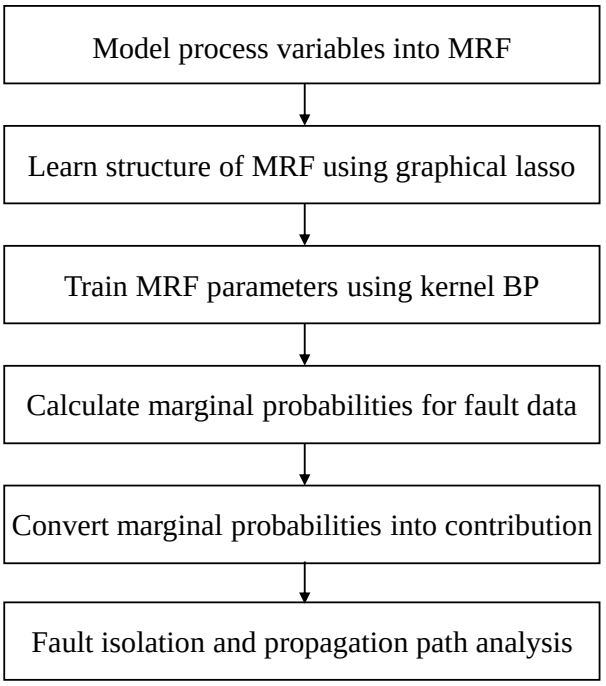


Figure 4.1: Sequence of fault isolation and propagation path analysis using MRF modelling and KBP.

4.3.1 MRF structure learning via graphical lasso

MRF structure learning of process variables incorporating the iterative use of the graphical lasso, is the same as shown in Sections 2.4 and 2.5.

4.3.2 Kernel belief propagation - bandwidth selection

After the structure of the MRF is learned using graphical lasso, the KBP algorithm is implemented to obtain the Gram matrices for each of the system variables, then conduct inference with respect to the fault data set. As is the case in the work by Song et al.[68], the Gaussian kernel is used in this study, since it has no difficulties expressing the nonlinearity encountered in process variables. The KBP algorithm is based on kernel trick calculations where the iterative message update calculation is done within the RKHS, and implements kernel density estimation for marginal belief calculation. During this process one the most important issues is the calculation of the bandwidth for each of the variables. Bandwidths are critical when implementing kernel calculations, since they can severly alter the feature of the underlying nature of the data. In this study, the optimal bandwidth for each of the variables are calculated using cross validation. In cross validation, the estimation model is fit to part of the data, then using a quantitative metric, such as the maximum likelihood, it is determined how well this models fits the remaining data. Thus the optimal bandwidth is calculated by iteratively calculating the maximum likelihood function. During belief calculation in Equation 4.3, the marginal probability, $P^{*}_{X_s}$, is calculated by kernel density estimation

of each of the variables. Thus, here the bandwidth values are calculated for each of the univariate density estimates. When performing message updates in Equation 4.10, the kernel calculation is done in a pairwise manner, since the KBP algorithm is based on pairwise MRFs. Thus in this case, separate bandwidth values obtained for every bivariate density estimation are used.

4.3.3 Conditional contribution evaluation

To conduct fault isolation and propagation path analysis, the condtional marginal probability values of the variables for each of the samples obtained from Equation 4.10, should be converted into contribution values. In this study, the conditional contribution of a variable X is defined as:

$$C(X) = log(\frac{1}{P(X)}) \tag{4.11}$$

where P is the conditional marginal probability computed for each variable in each data sample, and C is the conditional contribution value.

The conditional contribution is defined to have a inverse relationship with conditional marginal probability, since conditional contribution measures the likeliness of a variable being in a faulty condition, as opposed to conditional marginal probability. Logarithm is implemented to the inverted conditional marginal probability value since the multiplication of messages from neighboring nodes result in values with high orders of magnitude.

When using the conditional contribution for fault diagnosis, two factors should be focused on: the magnitude and the reaction speed, referring to how fast a certain variable shows a sudden increase in conditional contribution after the initiation of a fault. The root fault variable can be analyzed by detecting the variable with the largest contribution value. Also, the fault propagation path can be analyzed by following the chronical order of reaction speed for each of the variables. The fault can be observed to be propagating from the fast responding variables to the slow responding variables, since it is reasonable to assume that variables having a large contribution value early is affected by the fault before other variables having large contribution values at a later time. The effectiveness of the use of conditional contribution values are verified by application of the method to actual process models, as shown in the next section.

4.4 Application Results & Discussion

The proposed fault diagnosis methodology is applied to two different process models, to verify its performance. The method is first applied to the two tank model used in the work of Lindner et al.(2017) to show the step-by-step application of the method, and the fault diagnosis results are shown. Also, to show the performance of the method when applied to more complicated plants, it is applied to the widely used TEP model [1]. Since most of the recent fault detection and isolation methods have been readily applied to the TEP, it serves the purpose of performance comparison, and verifies the extensive functionality of the proposed methodology.

4.4.1 Two tank process

The two tank process is a Matlab Simulink model that was used in Lindner et al.(2017)[63] to verify the usability of causality methods. It is composed of two tanks in series, where the cold water flow and external steam flow are used to control the level and temperature of the two tanks. To maximize the effect of process faults on the monitored variables, the open-loop version of the model is used in this study. The flowsheet of the two tank model is shown in Figure 4.2.

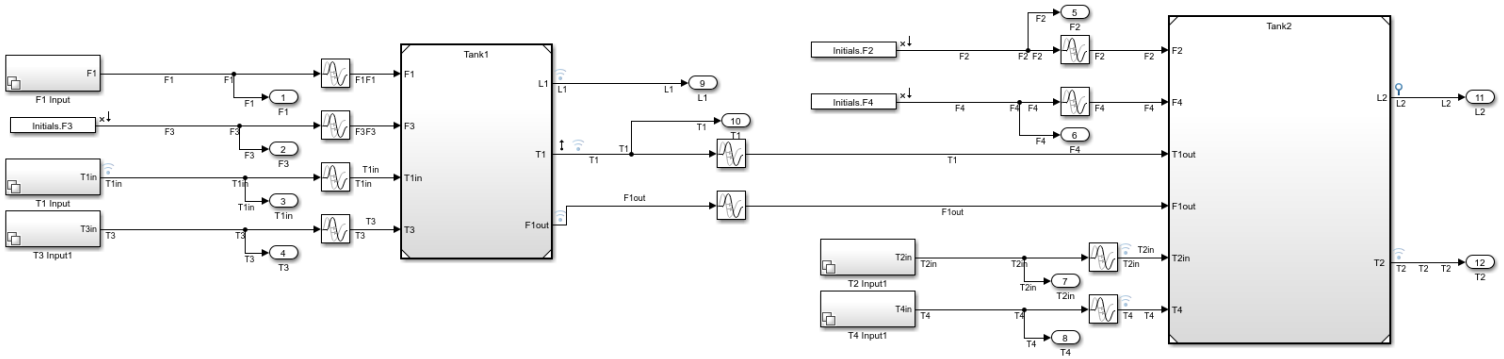


Figure 4.2: Simulink process flowsheet of the two tank model.

The two tank model consists of 12 process variables, which are shown in Table 4.1. Two types of faults, the step disturbance and the oscillating disturbance, can manually be implemented within the two tank model. In this study, five fault cases were generated on the five different variables that can be manipulated, which are shown in Table 4.2. Faults with step disturbances were designed to be initiated at the 1000^{th} sample point, and faults with oscillating disturbances were designed to be initiated at the start of the simulation.

The data from the 12 variables are sampled by 0.5 seconds. First the normal processing data without any process faults are extracted, and are used for training the MRF and obtaining the bandwidths of the variables. Then, process data for each of the five faults are obtained to conduct fault isolation and propagation path analysis.

The bandwidth values are obtained by maximimum likelihood calculation with respect to cross-validation, considering bivariate pairs of variables. This is to take into account the fact that the MRF of the process variables is modelled as a pairwise MRF. The bandwidth values for each variable, are given in Table 4.3. Since the bandwidth values are obtained according to normal process data, these values are used for fault diagnosis of all of the faults, when determining the contribution values.

Table 4.1: Monitored variables in the two tank process.

Process Variable	Description
F1	Cold water flowrate of tank 1
F3	Steam flowrate of tank 1
T1_in	Temperature of tank 1 cold water stream
T3	Temperature of tank 1 steam
F2	Cold water flowrate of tank 2
F4	Steam flowrate of tank 2
T2_in	Temperature of tank 2 cold water stream
T4	Temperature of tank 2 steam
L1	Tank 1 level
T1	Tank 1 temperature
L2	Tank 2 level
T2	Tank 2 temperature

Table 4.2: Five faults generated in the two tank process.

Process Variable	Fault type
F1	Step
T1	Oscillating
T3	Step
T2	Oscillating
T4	Step

Table 4.3: Bandwidth values of the process variables in the two tank process.

Process Variable	Bandwidth value
F_1	1.68e-4
F_3	4.49e-4
$T_{1,in}$	0.0229
T_3	0.0896
F_2	3.61e-5
F_4	3.52e-5
$T_{2,in}$	0.0225
T_4	0.0920
L_1	1.80e-3
T_1	0.0460
L_2	2.75e-3
T_2	0.0459

The conditional marginal probability of each of the variables, for every fault data sample, are calculated using KBP. The number of iterations required for convergence of belief propagation was determined as 20, which was sufficient for numerous test cases. To show the propagation of the fault to different variables, 200 samples starting from the 901^{st} point to the 1100^{th} point were evaluated. The varying contribution value is plotted as a image plot, where each pixel represents the conditional contribution value of a variable for a certain sample. For effective fault diagnosis the value of the condtional contribution for each sample was scaled with respect to the norm of the conditional contribution of that variable. To emphasize one of the results, the contribution analysis results for fault 1 is shown in Figure 4.3.

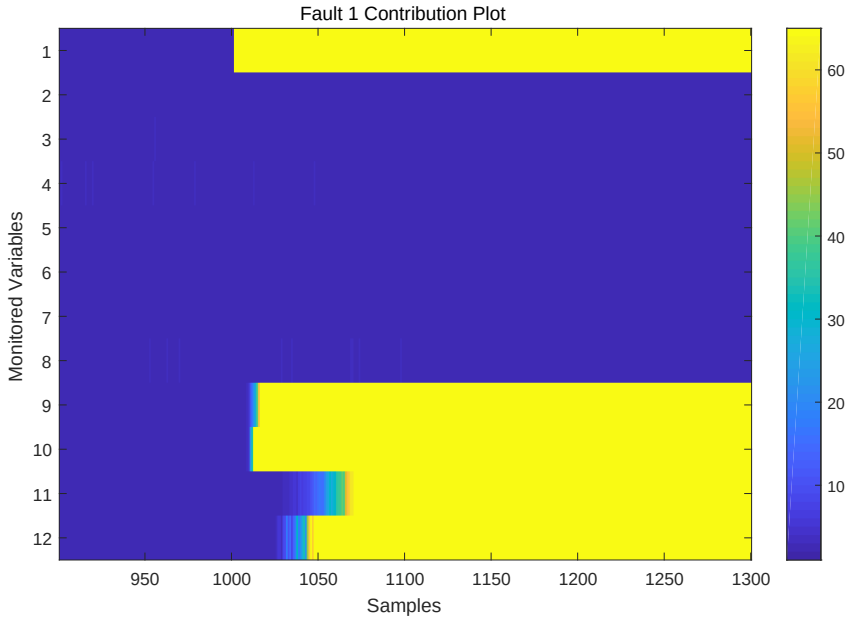
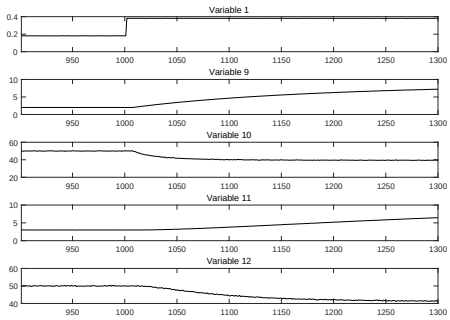


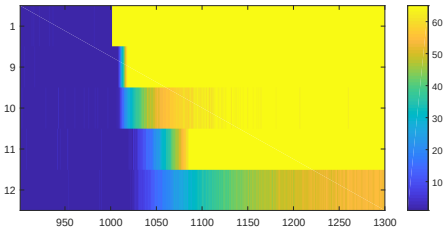
Figure 4.3: Contribution analysis results for fault 1.

According to the figure, the variables showing large conditional contribution values with respect to the fault are variables 1, 9, 10, 11, and 12. Also, the fault propagation path can be intuitively analyzed, observing the difference in reaction time for each of the variables. Variable 1, which is the F_1 variable, retains a high value from the start of the initiation of the fault, which is in accordace with the fact that F_1 is the root fault variable. Then, the fault affects variables 9 and 10, and then 11 and 12 sequentially. Variables 9, 10, 11, and 12 are the L_1, T_1, L_2 and T_2 variables, respectively. This is an expected result since evoking a step change in the input flow would alter the level and temperature of the two tanks, in sequential order. It is notable that the temperature variables T_1 and T_2 are affected slightly faster than level variables, which shows that the proposed methodology is sensitive enough to capture the characteristics of different variables.

The effectiveness of the contribution analysis is better emphasized by comparing its result with the original process data of the fault 1 case. The data flow plot and image plot of the five variables shown to be largely influenced by fault 1, variables 1, 9, 10, 11 and 12 are shown in Figure 4.4.



(a) Data flow plot



(b) Square plot

Figure 4.4: Data flow plot and square plot of variables 1, 9, 10, 11, and 12.

As can be seen in Figure 4.4a, the variation in the data flow of the variables is evident for the five variables, after the fault is initiated at variable 1. However, it is difficult to distinguish among variables the magnitude of variance inferred by the fault. While the root fault node may be evident due to the step changing form of variable 1, it is impossible to determine the fault propagation path, since the reaction speed and the magnitude of contribution cannot be determined. The reaction speed of each variable is more distinguishable in Figure 4.4b, since the magnitude of contribution is better visualized. The values shown in the graph are normalized process data values. Although it is possible to see the time difference of contribution increase for each variable, the order of reaction speed is different from that of Figure 4.3. This is because the values in Figure 4.4b do not take into account the message values coming in from different variables, thus not being able to adjust the contribution value with respect to the condition of other variables. This comparison shows that for precise isolation of the root node, and correct propagation path analysis, condtional contribution plot analysis is essential.

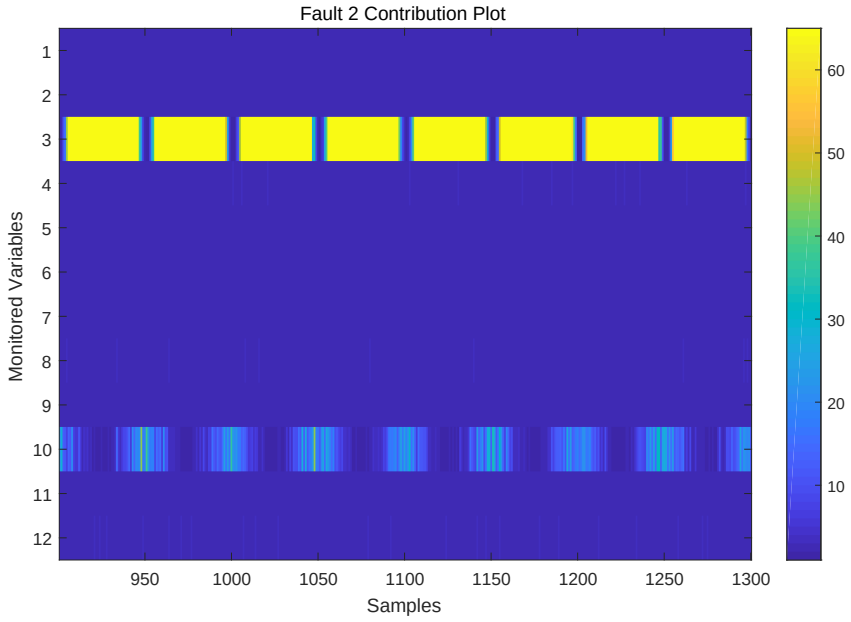


Figure 4.5: Contribution analysis results for fault 2.

The contribution analysis result for fault 2 is shown in Figure 4.5. Since fault 2 is an oscillatory fault, the conditional contribution plot shows a different trend compared to that of the fault 1 case, where the variables affected by the fault show oscillating contribution values. As for fault isolation, it can be determined intuitively that variable number 3 is the root fault node, since it shows the largest contribution value throughout the process run. Also, the propagation of fault effect is evident as well. The contribution values of variable 10 fluctuate along with the contribution value of variable 3, only with alternating peaks, and has a much smaller magnitude. Thus it can be deduced without difficulty that variable 10 is somewhere downstream of variable 3, and is affected by the fault initiated from variable 3.

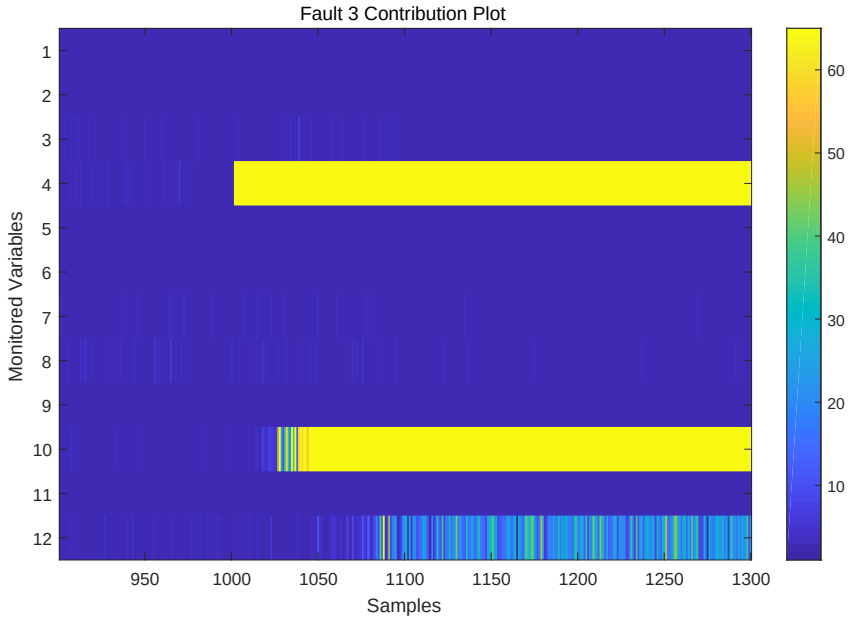


Figure 4.6: Contribution analysis results for fault 3.

The contribution analysis result for fault 3 is shown in Figure 4.6. The conditional contribution plot of fault 3 shows similar trends as that tof fault 1. Predicting from the magnitude of the contribution value and its reaction speed, fault 4 can be determined as the root fault node. Also, it shows that variable 10 is critically affected by the fault, shortly after its initiation on variable 4. Then variable 12 shows an increase in the contribution value, with a magnitude smaller than the values of variables 4 and 10. This is in accordance with the actual process analysis, since the temperature of the second tank is less affected by the increase in steam temperature of the first tank, due to the mitigation effect induced by the steam and cold water flow of the second tank. Also, this mitigation effect slows down the occurence of the fault effect on the variable, which is reflected in the delay of the reaction speed in variable 12.

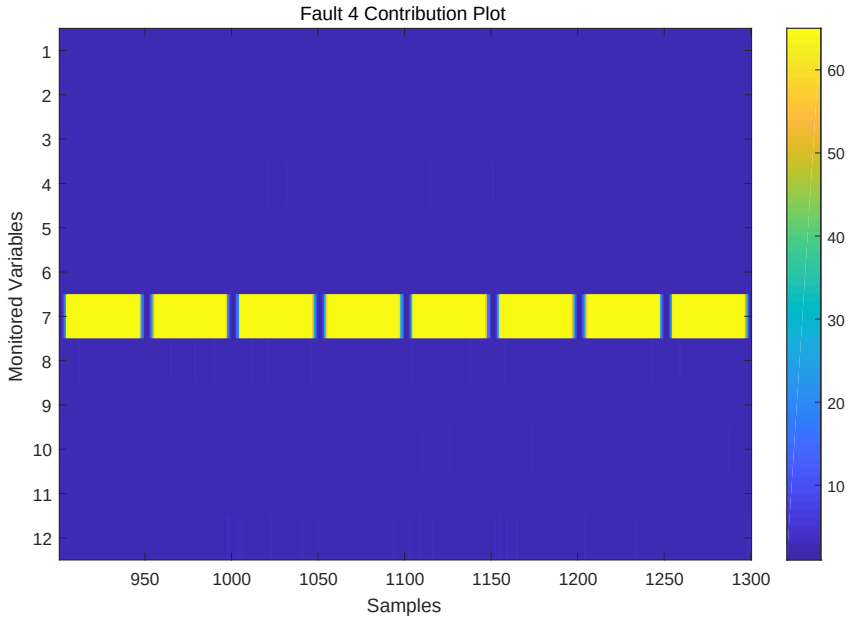


Figure 4.7: Contribution analysis results for fault 4.

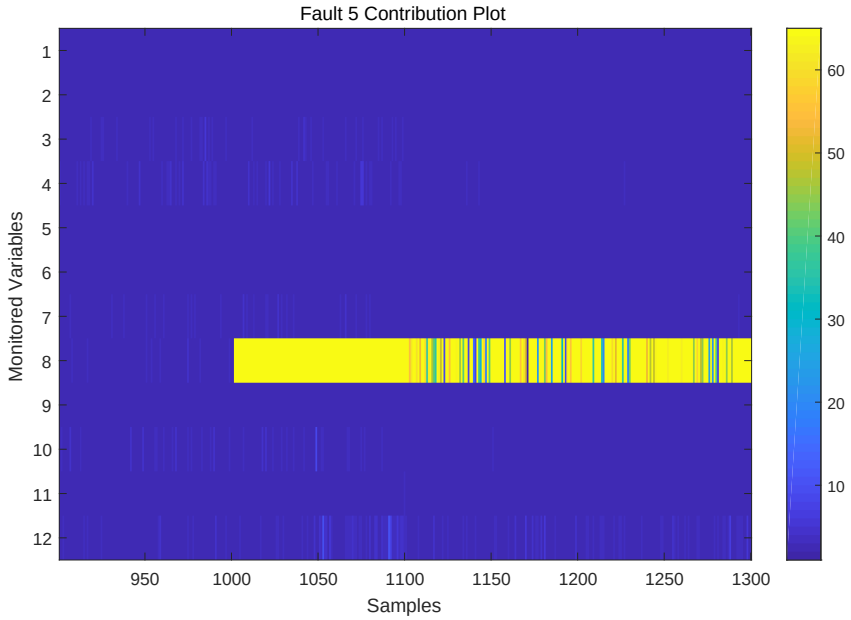


Figure 4.8: Contribution analysis results for fault 5.

The contribution analysis result for faults 4 and 5 are shown in Figure 4.7 and Figure 4.8, respectively. The conditional contribution value trend of fault 4 and 5 are similar to that of fault 2 and 3 respectively, since they share the characteristic of the fault (oscillatory and step increase, respectively). Thus, the root fault node can be determined without doubt, as the root variables (variable 7 for fault 4 and variable 8 for fault 5) show evidently high contribution values compared to other variables. However, although both the root nodes are not placed at the end of the process, no fault propagation is detected for both faults. This phenomenon can be explained by the fact that the root cause variables of faults 4 and 5 are both on the second tank. This means that although a temperature disturbance may occur on the cold water flow or the input steam flow of the second tank, the effect of the fault is greatly mitigated by the incoming flow from tank 1. As a result the temperature increment that should have appeared on the temperature of tank 2 is mitigated, thus showing no changes in the contribution plot. Since a step change acts as a more drastic disturbance than oscillatory changes, small changes in the contribution value of variable 12 can be seen in Figure 4.8, while no change is observed at all in Figure 4.7.

As can be observed from the contribution plots of the five faults in the two tank model, the contribution analysis provides excellent results in root fault variable isolation. Also, it shows effective fault propagation path analyzing capability, since the increase in contribution values for the nodes affected by the fault occurs in a timely manner.

4.4.2 Tennessee Eastman process

To verify the performance of the proposed method with a model more similar to an actual chemical process, the method was applied to the TEP. TEP is a widely used benchmark chemical process, consisting of five process units, the reactor, condenser, compressor, vapor/liquid separator, and the stripper. Four types of gaseous materials, A, C, D, and E, are put into the process to produce two types of products G and H, along with a by-product F. The process flowsheet of the TEP is shown in Fig 3.2.

When using the MATLAB version of TEP, the fault initiation time and duration of simulation can be user-defined, along with the type of fault to be applied. In this study, all of the faults were set to be introduced at the 1000^{th} data point, and the 21 simulations were run until the 7200^{th} data point to observe the change in process conditions.

When applying the proposed method to TEP it cannot be directly applied to the entire set of monitored variables, since obtaining the graph structure of 50 variables, and implementing the KBP algorithm on it is computationally infeasible. Thus, using the strategy proposed in our previous study (Kim et al.[55]), the TEP model is divided into separate groups, then the contribution of each group is analyzed separately for the different faults. The splitting of the monitored variables into separate groups are done by the iterative graphical lasso algorithm proposed in Kim et al.[55], which has shown good results with respect to process monitoring. The details of the variable grouping process is provided in Kim et al.[55].

Analogous to the sequence of the two tank model, each group of variables are mod-

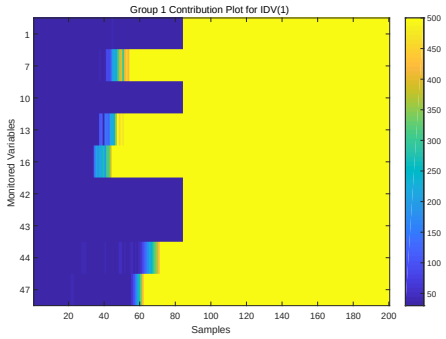
elled into pairwise MRFs, their structures are learned, and the conditional contribution is calculated for each variable using the KBP algorithm, separately for each group. Since the conditional contribution analysis process is done on separate groups, bandwidth values are calculated independently for each group as well. The bandwidth values of each variable when they are modelled into pairwise MRFs are shown in Table 4.4.

Table 4.4: Bandwidth values for each variable in the TEP model.

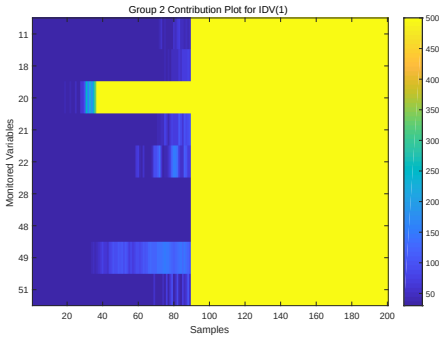
Group	No.	Bandwidth value	Group	No.	Bandwidth value
G1	1	0.0015	G3	35	0.0131
	7	0.2668		36	0.0131
	10	0.0027		40	0.1377
	13	0.2647		45	0.0233
	16	0.3082	G4	23	0.0687
	42	0.0300		24	0.0272
	43	0.0315		26	0.0256
	44	0.1436		27	0.0640
	47	0.3241		29	0.0684
G2	11	0.0186		31	0.0657
	18	0.0140		37	0.0027
	20	0.1033		38	0.0027
	21	0.0051		39	0.0026
	22	0.0404		41	0.1333
	28	0.0067	G5	2	4.8824
	48	0.0079		5	0.0520
	49	0.0050		6	0.0538
	51	0.0236		8	0.1283

G3	3	6.3106	9	0.0028
	4	0.0134	12	0.2614
	25	0.0640	14	0.0322
	30	0.0290	15	0.2640
	32	0.0252	17	0.0296
	33	0.0651	19	0.2866
	34	0.0067	52	0.4095

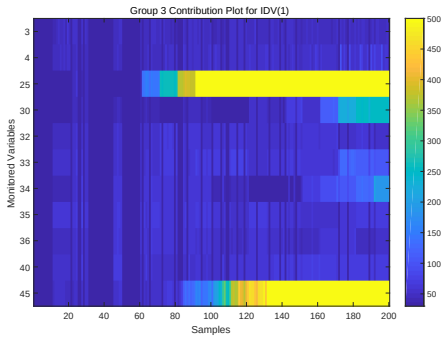
Using the bandwidth values, the conditional marginal probability of each of the variables, for each of the fault data samples, are calculated using the KBP algorithm. The same number of KBP iterations as the two-tank model are used, and the samples for conditional contribution calculation were from the 991^{th} point to the 1190^{th} point, to focus on the region of samples after the initiation of the fault. The resulting contribution plots for the case of the IDV(1) are shown in Figure 4.9. It should be noted that, although there are studies that analyze the root cause node of the TEP faults, such as the work by Bao et al.[10], Luo et al.[11], and [14], their results are incomparable to the results of the current study. This is because these previous studies implement the TEP model with restricted number of variables, selectively using variables that are suited for their purpose. In our study, all of the monitored process variables except the ones retaining constant values were selected for fault analysis to make the situation as similiar as possible to real-world applications.



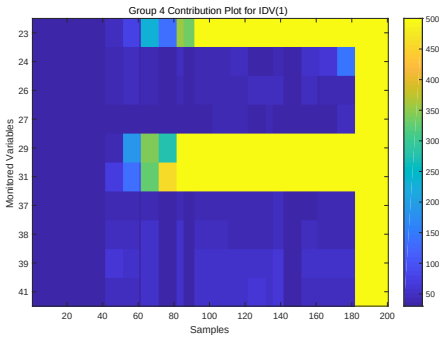
(a) Group 1



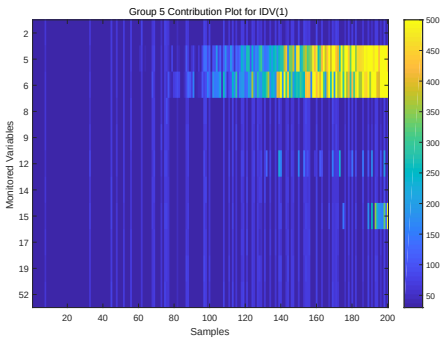
(b) Group 2



(c) Group 3



(d) Group 4



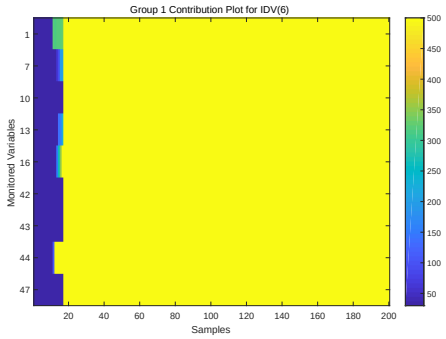
(e) Group 5

Figure 4.9: Contribution plots for IDV(1), of the five groups.

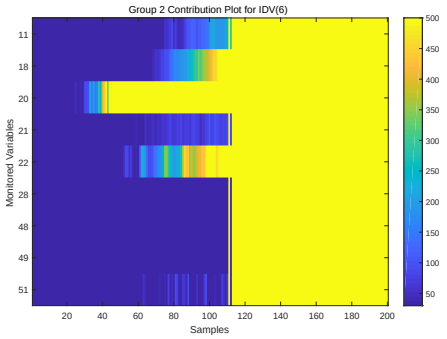
The resulting contribution plots show a more sophisticated form than the results obtained from the two-tank process. This could be due to the complex structure of the model including recycling streams, and the different sampling times of the composition related variables to other measured variables. This complexity requires analysis on the behavior of the implemented control structure as well as process changes to effectively analyze the fault propagation path, and to deduce the root fault node.

It should be noted that in most fault cases, except for IDV(6), IDV(14), IDV(15), and IDV(21), the variable directly related to the fault is not monitored[11]. Thus the variables that are closely related to, and are greatly affected by the fault would be detected as the root variable. For IDV(1), the variables most quickly reacting to the fault are distributed in groups 1 and 2, such as the variables 20, 16, 13, 7, 47, and 44. The quickest of these variables, variable 20, can be deemed as the root fault variable. Following the sequence of variable reaction speed, the order of fault propagation can be determined as: $20 \rightarrow 16 \rightarrow 13 \rightarrow 7 \rightarrow 47 \rightarrow 44$. Analyzing the identity of the designated variables, the root cause of the fault and the current situation of the process can be diagnosed. Intuitively it shows that the faulty condition is related to the pressure of the process, and that the compressor work is being adjusted to mitigate this change. Observing the large difference between the reaction time of variable 16 (stripper pressure) and 13 (separator), the fault is assumably induced from an altered condition within the stripper. This is advocated by the fact that variable 49, the manipulated variable for stripper liquid product flow, reacts very quickly to the fault. Also, as the

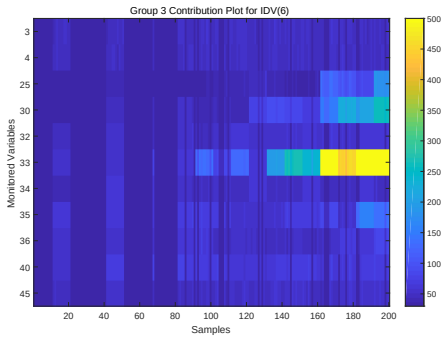
purge stream (variable 47) is being changed to react to this fault, the flow stream of A feed (variable 44) is variated as well, meaning that the fault is not related to an overflow in stream. Summing up from these observations, it can be reasonably deduced that the composition of the input stream to the stripper, is the starting point of the fault. By analyzing the later time region of the analyzed samples, the components critically related to the fault can be further specified. Other than the quickly reacting variables mentioned above, variables showing large contributions with fast reaction speed are 23, 25, 29, and 31. These are all composition variables of the reactor input stream and the purge stream. Moreover, it can be seen that these variables react with higher intensity at a much faster time period, compared to the other composition variables in the monitoring set, such as variables 24, 25-28, 30, and 32-36. Variables 23 and 25 are composition variables of A, and 29 and 31 are composition variables of C, clearly showing that the intitiated fault is related to the composition of these two specific components. Thus by observing the fault propagation sequence, and the identity of the variables with fast reaction and high contribution values, the specific characteristics of the fault can be diagnosed clearly.



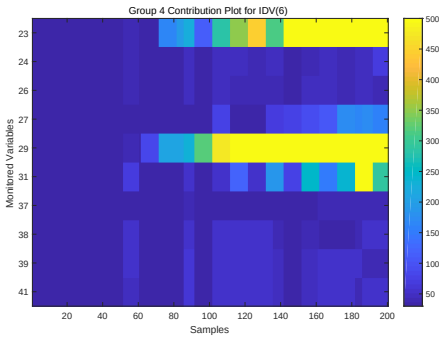
(a) Group 1



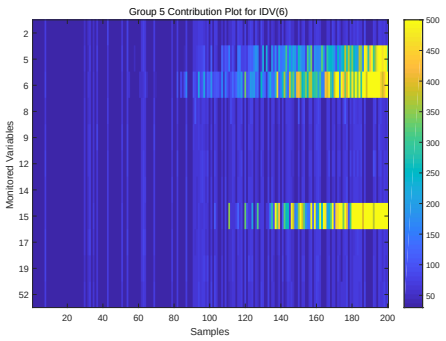
(b) Group 2



(c) Group 3



(d) Group 4



(e) Group 5

Figure 4.10: Contribution plots for IDV(6), of the five groups.

Another fault example, IDV(6), is tested out to provide analysis on a case where the specific fault variable is being monitored. The results are shown in Figure 4.10. The contribution plots of the five groups for IDV(6) shows different trends of results compared to IDV(1), showing that the list of variables affected by the fault are quite different. Likewise to IDV(1), the groups fastest to react to the initiation of the fault at the 10^{th} sample are groups 1 and 2. Especially, all of the nodes in group 1 show large conditional contribution values almost immediately after the initiation of the fault, and since variable 1 shows the fastest response, it can be readily determined as the root fault node, which is the case for IDV(6) where the A feed is lost in a stepwise manner. Variable 44 is the next quickest to respond to the fault, reacting almost simultaneously with variable 1, and showing high contribution values. This advocates the analysis that variable 1 is the fault node, since variable 44 is the manipulated variable node for A feed flow. Thus it can be presumed that a faulty condition has been produced in variable 1 (A feed flow), and that the control structure is working to mitigate this condition. Afterwards, analysis of rest of the variables in group 1 are straightforward. Variables related to the pressure of the process units and streams (variable 7, 10, 13, 16), as well as the manipulated variables related to the process streams show fast and strong contribution responses to the initiated fault, since flow and pressure effects are quickly propagated throughout the process. In group 2, the contribution value of variable 20 is similar to that of IDV(1), but some differences can be observed for variables 18, 22, and 49. In IDV(6), the variables 18 and 22 show more clear contribution responses to the

fault, whereas for IDV(1) these variables barely showed any contribution. On the other hand, the condtional contribution value of variable 49 is very small before retaining a high value near the 110^{th} sample, whereas in IDV(1) the contribution of variable 49 was a key evidence for determining the characteristic of IDV(1). This is due to the different characteristics of the faults. Temperature is more dynamically affected by change in stream flow, resulting in high contribution values of variables 18 and 22 in IDV(6). Composition changes have less affects to temperature compared to flow, which results in the contribution graph of IDV(1). This trend is observable in variables of groups 3 and 4 as well. Variables in groups 3 and 4 of IDV(6) show relatively small contributions compared to IDV(1), since they consist of mostly component related variables. Only conditional contributions for component A retains a high value, since the flow change of A stream affects the amount of component A as well. Variables in group 5 show similar trends for IDV(1) and IDV(6), since the largely affected variables are flow related variables (variables 5, 6, and 15), being altered either by composition disturbance or flow disturbance. Likewise in the case for IDV(1), the case of IDV(6) shows that the conditional contribution plot analysis allows specification of the root fault node, and that the characteristic of the fault can be analyzed, by observing variables with fast responses and large contribution values.

Analogous to the two fault cases, IDV(1) and IDV(6), the proposed fault diagnosis method can be used to isolate and analyze the root cause node and the propagation path of the rest of the faults programmed in the TEP. The fault diagnosis results for the 20

faults of TEP are summarized in Table 4.5. The second column in Table 4.5 shows the root cause variables and the propagation of the fault to other variables, determined in terms of the magnitude and reaction speed of the conditional contribution values. The numbers in parenthesis depict the variables that show identical contribution magnitude and reaction speed. The third column describes the analysis result of each fault, solely based on conditional contribution evaluation. The fault analysis results show that the location of the fault and its characteristics, including the intensity, can be determined via conditional contribution evaluation. For instance, comparing IDV(3) and IDV(9), the entire set of variables in group 4, which are composition variables, are disrupted for IDV(9), where IDV(3) only disrupts group 3 variables. This is in accordance with the variation amplitude of the reactor temperature variable (9) observed from the TEP model. However it is difficult to distinguish the specific fault locations of the temperature related faults, such as IDV(10) and IDV(16), since small changes in temperature only affects the composition variables throughout the process, rather than showing large disturbances on the specific variable. For IDV(13), where a fault occurs on the reaction kinetics, it is not possible to specify the characteristic of the fault, since the affects are only indirectly observable. However thorough analysis of the affected variables provides insight into the fault scheme, where the operator can determine it as a complex fault concerning both temperature and pressure disturbances.

Table 4.5: Fault diagnosis results for the 21 fault cases in the TEP.

Fault No.	Fault diagnosis	Fault analysis
IDV(1)	$20 \rightarrow 16 \rightarrow 13 \rightarrow 7 \rightarrow 47 \rightarrow 44$	Disturbance in A&C components in feed stream of stripper
IDV(2)	$(7, 13, 16) \rightarrow 47 \rightarrow 20 \rightarrow 30$	Disturbance in B component in feed stream of stripper
IDV(3)	(3,4,25,30,32,33,34,35,36,40,45)	Temperature disturbance
IDV(4)	$21 \rightarrow 51 \rightarrow 47 \rightarrow 9$	Reactor cooling water temperature disturbance
IDV(5)	$(3, 4, 25, 30, 32, 33, 34, 35, 36, 40, 45)$ $\rightarrow (23, 24, 26, 27, 29, 31, 37, 38, 39, 41)$	Temperature disturbance
IDV(6)	$1 \rightarrow 7 \rightarrow 13 \rightarrow 16 \rightarrow 44$	Flow disturbance in stream 1
IDV(7)	$(7, 13, 16, 4, 45)$ $\rightarrow (1, 10, 42, 43, 44, 47, 3, 25, 30, 32, 33, 34, 35, 36, 40)$	Pressure disturbance in feed stream of stripper

IDV(8)	$(3,4,25,30,32,33,34,35,36,40,45)$	Minor disturbance in stripper feed composition
	$\rightarrow (23,24,26,27,29,31,37,38,39,41) \rightarrow 20 \rightarrow 44$	
IDV(9)	$(3,4,25,30,32,33,34,35,36,40,45)$	Temperature disturbance (higher than IDV(3))
	$\rightarrow (23,24,26,27,29,31,37,38,39,41)$	
IDV(10)	$(3,4,25,30,32,33,34,35,36,40,45)$	Temperature disturbance
	$\rightarrow (23,24,26,27,29,31,37,38,39,41)$	
IDV(11)	$51 \rightarrow 9 \rightarrow 13 \rightarrow 16 \rightarrow 7 \rightarrow 21$	Reactor cooling water temperature disturbance
IDV(12)	$(3,4,25,30,32,33,34,35,36,40,45)$	Temperature disturbance
	$\rightarrow (23,24,26,27,29,31,37,38,39,41)$	
	$\rightarrow 13 \rightarrow 16 \rightarrow 7$	
IDV(13)	$(7,13,16,47) \rightarrow (1,10,42,43,44) \rightarrow 21$	Complex disturbance from upstream concerning
	$\rightarrow 51 \rightarrow 11 \rightarrow 22 \rightarrow 20 \rightarrow 30 \rightarrow 48$	temperature and pressure
IDV(14)	$51 \rightarrow 21 \rightarrow 9$	Reactor cooling water temperature disturbance

IDV(15)	$(3, 4, 25, 30, 32, 33, 34, 35, 36, 40, 45)$	Temperature disturbance
	$\rightarrow (23, 24, 26, 27, 29, 31, 37, 38, 39, 41)$	
IDV(16)	$(3, 4, 25, 30, 32, 33, 34, 35, 36, 40, 45)$	Temperature disturbance
IDV(17)	$21 \rightarrow 51$	Reactor cooling water temperature disturbance
IDV(18)	22	Condenser cooling water temperature disturbance
IDV(19)	$48 \rightarrow 49 \rightarrow 14$	Disturbance in separator bottom flow
IDV(20)	$13 \rightarrow 16$	Separator pressure disturbance
IDV(21)	$(3, 4, 25, 30, 32, 33, 34, 35, 36, 40, 45)$	Temperature disturbance
	$\rightarrow (23, 24, 26, 27, 29, 31, 37, 38, 39, 41)$	

Chapter 5

Concluding Remarks

In this thesis, a novel process monitoring framework for fault detection and diagnosis, the Glasso-MRF monitoring framework, is proposed. The methodology is to model the monitored process variables into Markov random fields, then group the variables into highly correlated sets, providing efficiency to the monitoring process. Fault detection is conducted using monitoring statistics based on kernel density estimation inference, and fault diagnosis is performed using conditional contribution values of the variables with respect to the fault, which is computed via the kernel belief propagation.

First, the process of modelling monitored variables into Markov random fields, obtaining the optimal division of groups for the variables and learning their structures is proposed. The graphical lasso algorithm is selected as the MRF structure learning algorithm. However, since the graphical lasso provides the structure of only a portion of the entire set of variables, the iterative graphical lasso algorithm is proposed so that the graphical lasso is applied to the remaining set of variables after each iteration,

until the desired number of variable groups are obtained. To set the number of variable groups, a sensitivity test on the regularization parameter ρ is carried out, observing the computational time and average fault detection accuracy provided by the different number of variables. Analysis results for applying the iterative graphical lasso algorithm on the Tennessee Eastman process showed that dividing the process variables into five groups is the most reasonable choice, since it requires reasonable computation time during fault detection and shows fault detection accuracy values of over 95%. Analyzing the variable members of each of the five groups obtained through iterative graphical lasso, it showed that the variables in each group are highly-correlated by various aspects, including thermodynamic properties, control structure relationships, and the geometrical location of the variable sensors. Each of the groups are individually subject to fault detection and diagnosis.

Secondly, a fault detection methodology based on the MRF groups and their structures obtained from iterative graphical lasso is proposed. The monitoring statistic is obtained by probability calculation of each of the variable groups through MRF inference. For fault detection inference the nonparametric algorithm of kernel density estimation is used, so that online monitoring is possible by bypassing the computation of the global normalization function, Z. Due to the curse of dimensionality, 70,000 normal processing data points are used to train the bandwidth parameters for kernel density estimation, for every variable in each subset. This does not affect the online monitoring speed and performance of the MRF monitoring since training is done of-

fline. The fault detection criterion for each subset is calculated by setting a false alarm rate of 5%, and faults are detected if the value of monitoring statistic is lower than this limit. Application of the method to the Tennessee Eastman process showed that the Glasso-MRF fault detection effectively detects various forms of process faults, showing a fault detection accuracy of over 95% for all of the 28 fault cases embedded within the Tennessee Eastman process. The significantly improved performance of the Glasso-MRF fault detection method is emphasized using the performance index, which is calculated by dividing the fault detection accuracy of other methods by that of the Glasso-MRF method. This improvement shows that dividing the monitored variables into highly-correlated groups affects the fault detection efficiency in terms of performance and speed, since the individual variable characteristics are preserved during the fault detection process.

Finally, a fault diagnosis methodology, including isolation of the root cause node of a fault, and analysis of the fault propagation path, is proposed. Using the MRF structures obtained from iterative graphical lasso, the MRFs are trained nonparametrically and the inference values for each faulty sample are calculated using the kernel belief propagation algorithm. Kernel belief propagation allows inference of MRFs by converting the message update process into linear operations within the reproducing kernel Hilbert space via conditional distribution embedding representations, while retaining reasonable computation time. When a fault occurs, the conditional marginal probability values of each variable is obtained through kernel belief propagation. These values are converted

into the conditional contribution, a newly defined value in this study as the logarithm of the inverse of conditional marginal probability. To examine the contribution of a variable with respect to a fault, the magnitude and the reaction speed of the conditional contribution of each variable is analyzed. The root cause node can be isolated by detecting the variable with the fastest response, then the fault propagation path can be determined by following the trail of value change of conditional contribution among the variables. Glasso-MRF fault diagnosis was applied to two industrial case studies, the two-tank process and the Tennessee Eastman process to verify its performance. Fault diagnosis results for 5 fault cases designed in the two-tank process, and for 28 fault cases embedded within the Tennessee Eastman process, all showed good diagnosis capabilities, where the root cause nodes are isolated even when they are not being monitored.

The Glasso-MRF provides a comprehensive framework for both fault detection and diagnosis. By modelling the process variables into MRFs, and applying the iterative graphical lasso and inference algorithms, the framework is able to monitor the process without neglecting individual characteristics of the variables, while firmly realizing variable relationships. The performance of the Glasso-MRF monitoring framework was verified by applying it to the Tennessee Eastman process, and it proved to be very competitive compared to other monitoring methods, correctly detecting and diagnosing all forms of complex process faults.

Future work would be to improve the computation time of the kernel belief propaga-

tion, and to improve the group number determination process during iterative graphical lasso applications. Kernel belief propagation computation can be improved using the approximate message update procedure suggested in Song et al. [68]. Determination of the number of sets for iterative graphical lasso could be automated by formulating the process into an optimization problem, where the objective function can be modelled as a combination of the computation time and fault detection accuracy.

Bibliography

[1] J.J. Downs and E.F. Vogel. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245 – 255, 1993.

[2] Ruben Gonzalez, Biao Huang, and Eric Lau. Process monitoring using kernel density estimation and Bayesian networking with an industrial case study. *ISA Transactions*, 58:330–347, 2015.

[3] Weiwu Yan, Pengju Guo, Liang gong, and Zukui Li. Nonlinear and robust statistical process monitoring based on variant autoencoders. *Chemometrics and Intelligent Laboratory Systems*, 158:31 – 40, 2016.

[4] Tiago J Rato and Marco S Reis. Fault detection in the Tennessee Eastman benchmark process using dynamic principal components analysis based on decor-related residuals (DPCA-DR). *Chemometrics and Intelligent Laboratory Systems*, 125:101–108, 2013.

[5] Shen Yin, Steven X. Ding, Adel Haghani, Haiyang Hao, and Ping Zhang. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, 22(9):1567 – 1581, 2012.

[6] Qiugang Lu, Benben Jiang, R. Bhushan Gopaluni, Philip D. Loewen, and

Richard D. Braatz. Sparse canonical variate analysis approach for process monitoring. *Journal of Process Control*, 71:90 – 102, 2018.

[7] David L Olson and Dursun Delen. *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[8] Andreas Bathelt, N Lawrence Ricker, and Mohieddine Jelali. Revision of the Tennessee Eastman process model. *IFAC-PapersOnLine*, 48(8):309–314, 2015.

[9] Leo H Chiang, Evan L Russell, and Richard D Braatz. *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[10] Shiyi Bao, Lijia Luo, Jianfeng Mao, and Di Tang. Improved fault detection and diagnosis using sparse global-local preserving projections. *Journal of Process Control*, 47:121–135, 2016.

[11] Lijia Luo, Shiyi Bao, Jianfeng Mao, and Di Tang. Fault detection and diagnosis based on sparse PCA and two-level contribution plots. *Industrial & Engineering Chemistry Research*, 56(1):225–240, 2016.

[12] Lijia Luo, Shiyi Bao, Jianfeng Mao, and Zhenyu Ding. Industrial process monitoring based on knowledge–data integrated sparse model and two-level deviation magnitude plots. *Industrial & Engineering Chemistry Research*, 57(2):611–622, 2018.

[13] Sylvain Verron, Jing Li, and Teodor Tiplica. Fault detection and isolation of faults

in a multivariate process with Bayesian network. *Journal of Process Control*, 20(8):902–911, 2010.

[14] Junichi Mori, Vladimir Mahalec, and Jie Yu. Identification of probabilistic graphical network model for root-cause diagnosis in industrial processes. *Computers & Chemical Engineering*, 71:171 – 209, 2014.

[15] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[16] Michael I Jordan. An introduction to probabilistic graphical models, 2003.

[17] Jinsong Zhao, Lin Cui, Lihua Zhao, Tong Qiu, and Bingzhen Chen. Learning HAZOP expert system by case-based reasoning and ontology. *Computers & Chemical Engineering*, 33(1):371 – 378, 2009.

[18] S. Dey and J.A. Stori. A Bayesian network approach to root cause diagnosis of process variations. *International Journal of Machine Tools and Manufacture*, 45(1):75 – 91, 2005.

[19] Zhengbing Yan and Yuan Yao. Process fault isolation via Bayesian lasso-based reconstruction analysis. In Antonio Espuña, Moisès Graells, and Luis Puigjaner, editors, *27th European Symposium on Computer Aided Process Engineering*, volume 40 of *Computer Aided Chemical Engineering*, pages 1669 – 1674. Elsevier, 2017.

[20] Sylvain Verron, Teodor Tiplica, and Abdessamad Kobi. Multivariate control charts with a Bayesian network. In *4th International Conference on Informatics in Control, Automation and Robotics (ICINCO'07)*, pages 228–233, Angers, France, 2007.

[21] Ramesh Vaidhyanathan and Venkat Venkatasubramanian. Digraph-based models for automated HAZOP analysis. *Reliability Engineering & System Safety*, 50(1):33 – 49, 1995.

[22] Manuel Rodríguez and José Luis de la Mata. Automating HAZOP studies using d-higraphs. *Computers & Chemical Engineering*, 45:102 – 113, 2012.

[23] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[24] Rajesh Talluri, Veerabhadran Baladandayuthapani, and Bani K Mallick. Bayesian sparse graphical models and their mixtures using lasso selection priors. *arXiv preprint arXiv:1310.1127*, 2013.

[25] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[26] Brenden Lake and Joshua Tenenbaum. Discovering structure by learning sparse graphs. *Cognitive Science Society, Inc.*, 2010.

[27] Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of

Markov networks using l_1-regularization. In *Advances in Neural Information Processing Systems*, pages 817–824, 2007.

[28] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Applications of the lasso and grouped lasso to the estimation of sparse graphical models. Technical report, Technical report, Stanford University, 2010.

[29] Rahul Mazumder and Trevor Hastie. The graphical lasso: New insights and alternatives. *Electronic Journal of Statistics*, 6:2125, 2012.

[30] Rahul Mazumder and Trevor Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13(Mar):781–794, 2012.

[31] Nicolai Meinshausen and Peter Buhlmann. High-dimensional graphs and variable selection with the lasso. *Ann. Statist.*, 34(3):1436–1462, 06 2006.

[32] Ryan Adams, Hanna Wallach, and Zoubin Ghahramani. Learning the structure of deep sparse graphical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 1–8, 2010.

[33] Onureena Banerjee, Laurent El Ghaoui, and Alexandre dąŕAspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine learning research*, 9(Mar):485–516, 2008.

[34] Jason D. Lee and Trevor J. Hastie. Learning the structure of mixed graphical

models. *Journal of Computational and Graphical Statistics*, 24(1):230–253, 2015. PMID: 26085782.

[35] Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R Nevins, Guang Yao, and Mike West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004.

[36] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[37] Qiao Liu, Xiao Ma, Weihua Ou, and Quan Zhou. Visual object tracking with online sample selection via lasso regularization. *Signal, Image and Video Processing*, 11(5):881–888, Jul 2017.

[38] Patricia Menéndez, Yiannis AI Kourmpetis, Cajo JF ter Braak, and Fred A van Eeuwijk. Gene regulatory networks from multifactorial perturbations using graphical lasso: application to the DREAM4 challenge. *PloS one*, 5(12):e14147, 2010.

[39] Bhavik R. Bakshi. Multiscale PCA with application to multivariate statistical process monitoring. *AIChE Journal*, 44(7):1596–1610.

[40] Wenfu Ku, Robert H. Storer, and Christos Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179 – 196, 1995. InCINC '94 Selected papers from the First International Chemometrics Internet Conference.

[41] Manabu Kano, Shinji Hasebe, Iori Hashimoto, and Hiromu Ohno. A new multivariate statistical process monitoring method using principal component analysis. *Computers & Chemical Engineering*, 25(7):1103 – 1113, 2001.

[42] Leo H Chiang, Evan L Russell, and Richard D Braatz. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 50(2):243 – 252, 2000.

[43] Jong-Min Lee, ChangKyoo Yoo, and In-Beum Lee. Statistical process monitoring with independent component analysis. *Journal of Process Control*, 14(5):467–485, 2004.

[44] Zhiqiang Ge and Zhihuan Song. Process monitoring based on independent component analysis- principal component analysis (ICA- PCA) and similarity factors. *Industrial & Engineering Chemistry Research*, 46(7):2054–2063, 2007.

[45] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

[46] Sang Wook Choi, Changkyu Lee, Jong-Min Lee, Jin Hyun Park, and In-Beum Lee. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 75(1):55–67, 2005.

[47] Jong-Min Lee, ChangKyoo Yoo, Sang Wook Choi, Peter A. Vanrolleghem, and

In-Beum Lee. Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 59(1):223 – 234, 2004.

[48] Shriram Gajjar, Murat Kulahci, and Ahmet Palazoglu. Real-time fault detection and diagnosis using sparse principal component analysis. *Journal of Process Control*, 67:112 – 128, 2018. Big Data: Data Science for Process Control and Operations.

[49] Leo H Chiang, Mark E Kotanchek, and Arthur K Kordon. Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Computers & Chemical engineering*, 28(8):1389–1401, 2004.

[50] Jiusun Zeng, Shihua Luo, Jinhui Cai, Uwe Kruger, and Lei Xie. Nonparametric density estimation of hierarchical probabilistic graph models for assumption-free monitoring. *Industrial & Engineering Chemistry Research*, 56(5):1278–1287, 2017.

[51] Fan Yang and Deyun Xiao. Progress in root cause and fault propagation analysis of large-scale industrial processes. *Journal of Control Science and Engineering*, 2012, 2012.

[52] Usama Ahmed, Daegeun Ha, Jinjoo An, Umer Zahid, and Chonghun Han. Fault propagation path estimation in NGL fractionation process using principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 162:73–82, 2017.

[53] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[54] N Lawrence Ricker. Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control*, 6(4):205–221, 1996.

[55] Changsoo Kim, Hodong Lee, Kyeongsu Kim, Younggeun Lee, and Won Bo Lee. Efficient process monitoring via the integrated use of Markov random fields learning and the graphical lasso. *Industrial & Engineering Chemistry Research*, 57(39):13144–13155, 2018.

[56] Leo H. Chiang and Richard D. Braatz. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and Intelligent Laboratory Systems*, 65(2):159 – 178, 2003.

[57] Tao Yuan and S. Joe Qin. Root cause diagnosis of plant-wide oscillations using Granger causality. *Journal of Process Control*, 24(2):450 – 459, 2014. ADCHEM 2012 Special Issue.

[58] Han-Sheng Chen, Zhengbing Yan, Yuan Yao, Tsai-Bang Huang, and Yi-Sern Wong. Systematic procedure for Granger-causality-based root cause diagnosis of chemical process faults. *Industrial & Engineering Chemistry Research*, 57(29):9500–9512, 2018.

[59] Katerina Schindlerova. Equivalence of Granger causality and transfer entropy: A generalization. *Applied Mathematical Sciences*, 5:3637 –3648, January 2011.

[60] Thomas Schreiber. Measuring information transfer. *Phys. Rev. Lett.*, 85:461–464, Jul 2000.

[61] M. Bauer, J. W. Cox, M. H. Caveness, J. J. Downs, and N. F. Thornhill. Finding the direction of disturbance propagation in a chemical process using transfer entropy. *IEEE Transactions on Control Systems Technology*, 15(1):12–21, Jan 2007.

[62] Brian Lindner, Lidia Auret, and Margret Bauer. Investigating the impact of perturbations in chemical processes on data-based causality analysis. Part 1: Defining desired performance of causality analysis techniques. *IFAC-PapersOnLine*, 50(1):3269 – 3274, 2017. 20th IFAC World Congress.

[63] Brian Lindner, Lidia Auret, and Margret Bauer. Investigating the impact of perturbations in chemical processes on data-based causality analysis. Part 2: Testing Granger causality and transfer entropy. *IFAC-PapersOnLine*, 50(1):3275 – 3280, 2017. 20th IFAC World Congress.

[64] Yidan Shu and Jinsong Zhao. Data-driven causal inference based on a modified transfer entropy. *Computers & Chemical Engineering*, 57:173 – 180, 2013. PSE-2012.

[65] Payman Hajihosseini, Karim Salahshoor, and Behzad Moshiri. Process fault isolation based on transfer entropy algorithm. *ISA Transactions*, 53(2):230 – 240, 2014.

[66] Sylvain Verron, Jing Li, and Teodor Tiplica. Fault detection and isolation of faults

in a multivariate process with Bayesian network. *Journal of Process Control*, 20(8):902 – 911, 2010.

[67] Le Song, Arthur Gretton, and Carlos Guestrin. Nonparametric tree graphical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 765–772, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[68] Le Song, Arthur Gretton, Danny Bickson, Yucheng Low, and Carlos Guestrin. Kernel belief propagation. *arXiv preprint arXiv:1105.5592*, 2011.

[69] Erik B Sudderth, Alexander T Ihler, Michael Isard, William T Freeman, and Alan S Willsky. Nonparametric belief propagation. *Communications of the ACM*, 53(10):95–103, 2010.

[70] Shenlong Wang, Alex Schwing, and Raquel Urtasun. Efficient inference of continuous Markov random fields with polynomial potentials. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 936–944. Curran Associates, Inc., 2014.

[71] Alexander Ihler and David McAllester. Particle belief propagation. In *Artificial Intelligence and Statistics*, pages 256–263, 2009.

[72] Koichiro Yamaguchi, Tamir Hazan, David McAllester, and Raquel Urtasun. Continuous Markov random fields for robust stereo estimation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 45–58, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[73] Jian Peng, Tamir Hazan, David McAllester, and Raquel Urtasun. Convex max-product algorithms for continuous mrfs with applications to protein folding. In *Proc. ICML*, 2011.

Nomenclature

FDA	Fault detection accuracy
FDD	Fault detection and diagnosis
FDR	Fault detection rate
ICA	Independent component analysis
IDV	Individual faults for Tennessee Eastman process
KPCA	Kernel principal component analysis
MRF	Markov random field
MV	Manipulated variable
PCA	Principal component analysis
PGM	Probabilistic graphical model
TEP	Tennessee Eastman process
α	False alarm rate
β_{ut}	Message updates
γ	Performance index
ϕ_{st}	Potential function
Z	Partition function
m_{ts}	Messages in belief propagation
$C(X)$	Conditional contribution for variable X

국문초록

공정 이상의 감지 및 진단 시스템은 안전한 공정 운영에 필수적인 부분이다. 이상 감지는 이상이 발생했을 경우 즉각적으로 이를 정확하게 감지하는 프로세스를 의미하며, 대표적인 방법으로는 주성분 분석 및 오토인코더를 활용한 감지 방법론이 있다. 이상 진단은 결함의 근본 원인이 되는 노드를 격리하고, 이상의 전파 경로를 탐지하여 이상의 특성을 식별하는 프로세스이다. 공정 이상의 감지 및 진단 방법론에는 모델 분석 방법론, 지식 기반 방법론 등의 다양한 방법론이 있지만, 공정에 대한 적용 가능성과 성능 측면에서 가장 유용하다고 알려져 있는 데이터 기반 방법론이 널리 활용되고 있다. 공정 이상의 감지 및 진단에 대한 데이터 기반 방법론은 다방면으로 연구되어 왔지만, 이상 감지 및 진단을 모두 효과적으로 수행할 수 있는 방법론은 소수에 불과하며, 존재하고 있는 방법론들 역시 두 분야 모두에서 좋은 성능을 보여주고 있는 경우는 없다. 이는 기존 방법론들의 적용 가능성이 제한되어 있으며 공정에 적용시 제한된 성능을 보여주기 때문이다. 이상 감지의 경우, 대용량의 데이터를 처리할 때 발생하는 과부하로 인한 감지 능력의 저하, 차원 축소 방법론들을 사용할 시 이에 따른 변수 특성 반영의 부정확성, 그리고 축소된 차원에서의 계산으로 인하여 복합적인 형태의 이상을 감지해 내지 못하는 문제 등이 있다. 이상 진단의 경우 이상의 원인이 되는 노드의 격리 및 이상 전파 경로에 대한 분석이 부정확한 경우가 많은데, 이는 차원 축소로 인하여 공정 변수의 특성이 소실되는 성질이 있고, 방향성 그래프를 활용할 시 공정에 대한 선행 지식을 적용함으로써 편향된 이상

진단 결과가 나타나는 경우들이 발생하기 때문이다. 기존 방법론들에 대한 이러한 한계점들을 고려해 봤을때, 변수 각각의 특성이 소실되지 않도록하여 효과적으로 이상에 대한 감지와 진단을 모두 수행해 낼 수 있으면서도, 계산상의 효율성을 갖춘, 이상 감지 및 진단에 대한 통합된 방법론의 개발이 시급하다고 할 수 있다.

본 연구에서는 마르코프 랜덤 필드 모델링과 그래프 라쏘를 기반으로하여, 이상에 대한 감지 및 진단을 모두 수행해 낼 수 있는 통합적인 공정 모니터링 방법론을 제안한다. 마르코프 랜덤 필드는 비선형적이고 비정규적인 변수 관계를 효과적으로 모델링할 수 있게 해주고, 이상 발생 상황에서의 모니터링 통계값 계산시에 각 변수의 특성을 반영하여 확률 계산을 해 낼 수 있기 때문에 효과적인 이상 감지 및 진단 수단이 된다. 기본적으로 마르코프 랜덤 필드는 확률값 계산시의 부하가 크지만, 본 연구에서는 그래프 라쏘 방법론을 추가적으로 함께 활용하여 계산 상의 부하를 줄이고 효율적으로 이상 감지 및 진단을 해낼 수 있도록 하였다. 본 연구에서 제안된 내용들은 다음과 같다.

첫째, 공정 변수를 마르코프 랜덤 필드 형태로 모델링하고, 그래프 라쏘를 활용해 마르코프 랜덤 필드의 구조를 얻을 수 있는 방법론을 제시하였다. 그래프 라쏘는 마르코프 랜덤 필드의 구조를 파악하기 위한 방법론인데, 변수 간의 관계를 가우스 함수의 형태로 가정하기 때문에 다변수 시스템에서도 효율적으로 그래프 구조를 파악할 수 있도록 해준다. 본 연구에서는 반복적 그래프 라쏘를 제안하여 모든 공정 변수들이 상관관계가 높은 변수 집단으로 묶일 수 있도록 하였다. 이를 활용하면 전체 공정 변수 집단을 다수의 소집단으로 분류하고 각각에 대한 그래프 구조를 파악할 수 있게 되는데, 크게 두 가지의 효과를 얻을 수 있다. 우선적으로 마르코프

랜덤 필드 확률 계산의 대상이 되는 변수의 개수를 줄여줌으로써 계산 부하를 줄이고 효율적인 이상 감지가 이루어질 수 있도록 한다. 또한 상관관계가 높은 집단끼리 묶여서 모델링 된 그래프를 활용하여 이상의 진단 과정에서 공정 변수 간의 관계 파악 및 전파 경로 분석을 용이하도록 해준다.

두 번째로, 마르코프 랜덤 필드의 확률 추론을 기반으로 하여 효과적으로 이상 감지가 이루어질 수 있도록 하는 방법론을 제안하였다. 반복적 그래프 라쏘를 통해 얻어진 다수의 변수 소집단에 대하여 각각 확률 추론을 적용하여 이상 감지를 진행하게 되는데, 제안된 방법론에서는 커널 밀도 추정 방법론을 활용하였다. 정상 데이터를 활용하여 각 변수들에 대한 커널 밀도의 대역폭을 학습하고, 이상 데이터가 발생할 시 이를 활용한 커널 밀도 추정법을 사용하여 이상감시 통계치를 계산하게 된다. 이때 허위 진단율을 5%로 가정하여 각각의 소집단에 대한 공정 감지 기준선을 설정하였고, 이상감시 통계치가 공정 감시 기준선보다 낮게 될 경우 이상이 감지된다.

세 번째로, 이상 발생 시 원인이 되는 변수의 격리 및 이상 전파 경로 분석을 효과적으로 수행할 수 있는 방법론을 제시하였다. 제시된 방법론에서는 마르코프 랜덤 필드의 확률 추론 과정을 활용하여 이상 발생 시 각 변수의 조건부 한계 확률을 계산하고, 이를 활용해 새롭게 정의된 조건부 기여도 값을 계산하여, 이상에 대한 각 변수의 기여도를 파악할 수 있도록 한다. 이 과정에서는 커널 신뢰도 전파 방법론이 사용되는데, 이는 연속 변수를 가지는 마르코프 랜덤 필드에 대하여 확률 추론을 수행할 수 있도록 하는 방법론이다. 커널 신뢰도 전파법을 사용하면 정상 상태의 공정 데이터를 활용하여 마르코프 랜덤 필드를 구성하는 파라미터 값들을 학습하고, 이

상 발생시 이상 데이터에 대하여 각 변수의 조건부 기여도 값을 계산할 수 있게 된다. 이 때 계산된 조건부 기여도 값의 크기와, 이상 발생 이후 각 변수의 조건부 기여도 값의 변화 반응 속도를 종합적으로 판단하여, 이상의 원인 변수에 대한 격리와 이상 전파 경로 분석을 효과적으로 수행할 수 있게 된다.

본 연구에서는 제안된 이상 감지 및 진단 방법론의 성능을 검증하기 위하여 테네시 이스트만 공정 모델에 이를 적용하고 결과를 분석하였다. 테네시 이스트만 공정은 수년간 공정 감시 방법론을 검증하기 위한 벤치마크 공정으로 널리 사용되어 왔기 때문에, 제시된 방법론을 이에 적용해 봄으로써 다른 공정 감시 방법론들과의 성능을 비교해 볼 수 있었다. 또한 다수의 단위 공정을 포함하고 있고, 순환적인 변수 관계 역시 포함하고 있기 때문에 제시된 방법론의 성능을 시험해 보기에 적합했다. 테네시 이스트만 공정 내부에는 28개 종류의 이상이 프로그램 상에 내장되어 있는데, 제시된 공정 감지 방법론을 적용한 결과 모든 이상에 대하여 96% 이상의 높은 이상 감지율을 나타내었다. 이는 기존에 제시된 공정 감시 방법론들에 비하여 월등히 높은 수치였다. 또한 이상 진단 성능을 분석해 본 결과, 모든 이상에 대하여 원인이 되는 노드를 효과적으로 파악할 수 있었고, 이상 전파 경로 역시 정확하게 탐지하여 기존 방법론들과는 차별화된 성능을 나타내었다. 제시된 방법론을 테네시 이스트만 공정에 적용해 봄으로써, 본 연구 내용이 공정 이상의 감지 및 진단에 대한 통합적인 방법론 중에서 가장 우수한 성능을 나타내는 것을 확인할 수 있었다.

주요어: 공정 모니터링; 이상 감지 및 진단; 이상 전파 경로; 마르코프 랜덤 필드; 그래프 라쏘; 커널 신뢰 전파

학번: 2014-21514